Guidelines for Developing an Interactive Multimedia Prototype

Based on comparison of Low-and-High-fidelity prototypes in usability testing

J O H A N  L U N D B E R G

Master of Science Thesis
Stockholm, Sweden 2010
Guidelines for Developing an Interactive Multimedia Prototype

Based on comparison of Low-and-High-fidelity prototypes in usability testing

JOHAN LUNDBERG

Master’s Thesis in Media Technology (30 ECTS credits) at the School of Media Technology Royal Institute of Technology year 2010
Supervisor at CSC was Björn Hedin
Examiner was Nils Enlund

TRITA-CSC-E 2010:005
ISRN-KTH/CSC/E--10/005--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.csc.kth.se
Guidelines for Developing an Interactive Multimedia Prototype
- based on comparison of Low-and High-fidelity prototypes in usability testing

Abstract
The master thesis work has been carried out in Masters of Science in Media Technology at KTH as a mission from the PDU SD&P unit of Ericsson AB in Gothenburg. The implementation period was spent on this unit in Gothenburg and in close cooperation with Patrick Thomsson for creation of a prototype. The thesis partly extracts guidelines for development of interactive multimedia prototype and partly examines the importance of a prototype’s fidelity level, in the context of usability testing, under the condition where the low-fidelity is created as a copy from the high-fidelity prototype. Implementation and evaluation of two prototypes of Quick Builder (low- and high-fidelity) have been used as a basis for the compilation of guidelines, and the study of importance of a prototype’s fidelity in usability tests.

The purpose has been to compile guidelines for development of interactive prototypes in the multimedia area. Thus, the ambition is to give advice for future development of prototypes in the area, and localize increase of profits and pitfalls; based on literature study and experience from development and evaluation of the prototypes of Quick Builder.

The master thesis started in Gothenburg, compiling requirement specification of the prototype. This was followed by a study period within the prototype area and programming technology area. Thereafter the implementation was organized and scheduled, followed by the implementation of the high-fidelity prototype. When the prototype was done it was demonstrated in Kista outside Stockholm. After that the low-fidelity prototype was created as a copy of the implemented high-fidelity prototype. Both prototypes were evaluated in usability tests and finally compared by efficiency in finding usability problems. The whole process finally formed the guidelines.

The main conclusion indicates that there is no significant difference between the number of errors users made in the usability tests of prototypes of different level of fidelity, albeit statistically questionable, due to the low number of test participants. Nevertheless, there are tendencies indicating that different types of usability problems were discovered in the two prototypes. The high-fidelity prototype identified problems of higher severity compared to the low-fidelity prototype.
Riktlinjer för att utveckla en interaktiv multimediaprototyp
- baserat på jämförelse mellan låg- och hög-realistiska prototyper i användbarhets tester

Sammanfattning
Examensarbetet har utförts inom civilingenjörsutbildningen medieteknik på KTH och på uppdrag av Ericsson ABs enhet PDU SD&P i Göteborg. Under utvecklingsperioden tillbringades arbetet på PDU SD&P i Göteborg och i nära samarbete med Patrick Thomsson för framställning av en prototyp. Examensarbetet tar dels fram riktlinjer för utveckling av interaktiva multimediaprototyper och dels undersöker vikten av en prototyps naturtrogenhet (fidelity) i användbarhets tester, under villkoret att prototypen av låg naturtrogenhet är skapad utifrån prototypen av hög naturtrogenhet. Utveckling och utvärdering av två prototyper av Quick Builder (med låg- respektive hög naturtrogenhet) har legat som underlag för framställningen av riktlinjerna, samt undersökningen om vikten av en prototyps naturtrogenhet i användbarhets tester.

Syftet har varit att ta fram riktlinjer för utveckling av interaktiva prototyper inom området för multimedia. Detta med ambition för rådgivning av framtida utveckling av prototyper inom området, samt peka på effektivitetsvinster och fallgropar; baserat på litteraturstudier och erfarenheter från utvecklingen och utvärderingen av prototyperna av Quick Builder.


Den huvudsakliga slutsatsen är att det inte är någon märkbar skillnad på antalet fel som användare gör i användbarhets tester av prototyper av olika naturtrogenhet, om än något statistiskt osäkert, ty det låga antalet testdeltagare. Likväl finns det tendenser som tyder på att olika typer av användbarhetsproblem upptäcktes i de två prototyperna. Den prototypen av hög naturtrogenhet fann allvarligare användbarhetsproblem än prototypen av låg naturtrogenhet.
# Table of Contents

1. **Introduction** ................................................................. 5  
   1.1 Background ........................................................................ 5  
      1.1.1 Thesis background .................................................... 5  
      1.1.2 Company background .............................................. 5  
   1.2 Problem description .......................................................... 5  
   1.3 Goal and purpose ............................................................... 6  
   1.4 Question and hypothesis .................................................... 6  
   1.5 Approach ........................................................................ 7  
   1.6 Delimitation ................................................................... 8  

2. **Quick Builder prototypes** .................................................. 9  
   2.1 Requirement ...................................................................... 9  
   2.2 Goals for Quick Builder .................................................... 9  
   2.3 Creation of high-fidelity prototype ...................................... 10  
      2.3.1 The programming language ....................................... 10  
      2.3.2 Implementation ....................................................... 10  
   2.4 Creation of low-fidelity prototype ....................................... 15  

3. **Theory** .......................................................................... 16  
   3.1 Prototyping ......................................................................... 16  
      3.1.1 Different fidelities of prototype ................................. 16  
   3.2 Statistical analysis ............................................................. 18  
      3.2.1 Hypothesis testing ..................................................... 18  
      3.2.2 Welch’s $t$-test ......................................................... 18  
   3.3 Related work .................................................................... 19  

4. **Methodology** ................................................................ 21  
   4.1 Structure of study ............................................................. 21  
   4.2 Literature and technology search ....................................... 21  
   4.3 Usability test .................................................................... 22  
      4.3.1 Test user ................................................................. 22  
      4.3.2 Test environment and equipment .......................... 23  
      4.3.3 Description of test procedure ............................... 23  
   4.4 Questionnaire .................................................................. 24  
   4.5 Reliability and validity ...................................................... 24  
      4.5.1 Reliability ............................................................... 24  
      4.5.2 Validity ................................................................. 25  

5. **Results** ....................................................................... 26  
   5.1 Results from user tests ...................................................... 26  
   5.2 Results of shared usability errors ....................................... 28  
   5.3 Results from questionnaires ............................................. 29  

6. **Analysis** ....................................................................... 32  
   6.1 Comparison of results from the usability tests .................. 32  

7. **Conclusion** ................................................................... 34  
   7.1 Importance of fidelity level ............................................... 34
1. Introduction

This chapter describes two types of backgrounds, problem description, goal and purpose. Followed by questions, delimitations and what approaches that have been used.

1.1 Background

1.1.1 Thesis background
This thesis is requested by the telecom company Ericsson and was initially redesigned as a part of the former thesis Creation of Web Development Application, which was originally designed for two students. According to directions from the CSC (Computer Science and Communication) institution at KTH thesis shall be done individually. However two students are permitted to work for the same job initiator and with close related tasks as long as they produce their own report and presentation. This is what happened to this thesis project. The two-student-designed thesis where cut in half and provided with two different focuses. Although the thesis project was split into halves, there were moments when collaboration was required due to overlapping thesis coverage. The second student is mentioned as the collaborator and shows up in different chapters in this thesis.

1.1.2 Company background
There is a unit at Ericsson, named SD&P (Service Delivery and Provisioning), that has developed a complete business support system called MSDP (Mobile Service Delivery Platform). The MSDP is a package including a set of applications with different scopes of use. Portal Composer is one of them, a tool for creating mobile web sites, also known as mobile portals. It is popular among the personnel and known to be an advanced system that provides a wide range of functionality. Portal Composer is a powerful application that provides wide range of functions. The users can build the mobile site they want, the possibilities are many. The downside is that Ericsson’s average customer finds this application too complicated to handle in an effective way.

The MSDP package also contains Design Tool, a graphic design tool for portals, and Report Viewer for click statistics and ad repository usage reports. Finally, 3PI Manager which offers third party integration of content, such as ringtones, music, games and other suitable services for portals.

Most customers are requesting a simpler application supplied with the most commonly used tools, just enough for creating and maintaining a mobile web site. This kind of customer often has a profession like journalist or administrator with a limited level of technology knowledge. For them it would be enough to work with the basic components like inserting images and editing text, fonts and colors.

1.2 Problem description
Ericsson wants to enhance Portal Composer by developing an additional, but simpler application as a complement, called Quick Builder. To test the idea and reduce costs and resources, a first interactive prototype of Quick Builder shall be implemented. The implementation is one part of this thesis.

The SD&P usually makes a paper-based GUI sketch and then evaluates it. After approval a designer creates a digital copy at the computer that later is realized by programmers. The
problem is that they are not sure to what level it is profitable to develop a prototype before creating the real product.

This problem among others is put into questions listed below:

- What are the benefits from creating a prototype?
- To what level is it profitable to develop a prototype before creating the real product?
- Is it recommended to reuse code from a prototype in a real product?
- How shall key functions be selected?

This list forms the basis for the guidelines compiled in this thesis.

1.3 Goal and purpose

There are three goals to be reached in this thesis. The second and third goals are dependent on the first goal and cannot be reached until the first goal has been accomplished.

- Develop and evaluate a prototype of Quick Builder, an interactive prototype of a web based application for creation of mobile web sites.

- Write guidelines for developing interactive prototypes based on experience from developing Quick Builder, its evaluation results and studies. The purpose of these guidelines is to bring knowledge and advice for future prototyping in the multimedia area to make the development process more efficient in matter of time and cost.

- Find what importance a prototype’s level of fidelity has for the usability test and also test the hypothesis that there is no importance of the fidelity.

The intention of this goal is to get a deeper understanding of the guideline question, *To what level is it profitable to develop a prototype before creating the real product?*, by highlighting it from a different perspective.

1.4 Question and hypothesis

This thesis considers the following two questions:

- *What guidelines for developing an interactive multimedia prototype can be extracted from the implementation and evaluation of the Quick Builder prototypes?*

- *What is the importance of the prototype’s fidelity level, in the context of usability testing, under the condition where the low-fidelity is created as a copy from the high-fidelity prototype?*

The hypothesis that has been tested is:

- *There is no statistical difference between the mean errors the two groups made while testing the low- and high-fidelity prototype.*
1.5 Approach

This approach section intends to give a short overview of the thesis process, see Figure 1.1.

![Diagram](image.png)

Figure 1.1: Process for the prototypes of Quick Builder

Green blocks indicate cooperation between me and my collaborator, unfilled blocks indicate my collaborator’s steps and the filled blue blocks indicate my steps.

Firstly requirements analysis was compiled, consisted of information of user needs and Quick Builder goals.

Then a study was made to gather a whole picture of the prototyping area and different evaluation methods. Shortly after, a technology study was initiated. That provided a group of program languages with potentials of being selected for designing the prototype of Quick Builder. Program languages were chosen and a first paper sketch of Quick Builder was received by Ericsson.

Next, a moderate time of planning and structuring was done before I and my collaborator together started to develop the prototype, the implementation period began.

When the implementation was completed an interactive computer-based prototype stood ready for evaluation, the high-fidelity prototype. From this prototype a hand-drawn copy was created, the low-fidelity prototype. Both prototypes were evaluated with usability test and questionnaire, wherein the low-fidelity copy was evaluated by me and the other by my collaborator Patrick Thomsson.

Both evaluation results were compared and statistically analyzed. Unfortunately the evaluation was unable to produce statistically significant results for the differences between the low- and high-fidelity prototypes, due to the low number of test participants.

Finally, guidelines were compiled as a result from the implementation process from both fidelities and the evaluation results.
1.6 Delimitation

Due to lack of time a few areas have been excluded during the development of the interactive prototype. There is no documentation written for the prototype and the implemented features have a minimal underlying functionality. It has been taken minimal consideration to how the mobile websites (created inside the prototypes) work and look. There is almost none focus on design patterns for designing mobile websites or other recommendations for adapting websites to mobile phones.

The usability tests and the questionnaires only intend to evaluate the user interface. Usability testing is used as an evaluation tool in this thesis, but a focus on the method as such is omitted.

One aim of this thesis is to assemble guidelines for interactive prototyping within the multimedia area. These shall not be seen as the definite guidelines suitable for all projects, since they are based upon a case study where Quick Builder is the studied object. Nor will they give step by step instructions for how to make a prototype, but rather a few advices when the prototyping is coming to a point of decision, like “I have a product concept in mind, should I start creating a low-fidelity prototype or a high-fidelity prototype?”.

As mentioned in the section of thesis background and as the process illustration above shows, this thesis intersects with the collaborator’s thesis in some areas. What he did in these moments, where collaboration occurred, are described very briefly and may be fully described in his thesis.
2. Quick Builder prototypes

This chapter consists of four sections. Firstly there are requirements from customers. Next, goals for graphical user interface and lastly two sections about the creation of both fidelities. The section of high-fidelity prototype includes programming language and implementation. The last section is about how the low-fidelity prototype was created.

2.1 Requirement

A list of what the customer wants and needs from an application is called requirements. It is wise to have the requirements in mind when designing the application since it is the customer who will spend some time with it. This section lists customers’ requirements for Quick Builder provided by Ericsson:

One application: The last chapter introduced the problem of having unnecessary complex applications for creating the just simplest thing on a mobile portal. For those customers that would be pleased with the most used and basic functions, should have an easier application than Portal Composer, Report Viewer, Design Tool and 3PI Manager altogether. They want one application, not that cover all functionality from the four applications just mentioned, but just the most used ones to their everyday use.

Easy to learn: One application cannot solve every problem. Even though the right set of tools is available somewhere in the application it can be too tricky to find each of them. Especially when merging several applications into one, there is an imminent risk the groupings of different tools become too sprawling. The inconsistent look can possibly cause confusion and never gives the user a fair chance to learn the structure of the application. They want the application to be easy to learn.

Easy to use: As easy as the application should be to learn as easy should it be to use. Speaking the customer’s language is essential in making an easy-to-use application. For instance, embedding an audio clip with important information on the official webpage for deaf people is likely not the smartest approach for conveying a message. In this case the target audience has not been considered at all, which is not very successful. Good design gears to users’ needs. Logical navigation is another quality in good design as well as consistent layout of images, text, etc.

2.2 Goals for Quick Builder

Distinguish goals for Quick Builder from goals for the thesis, as presented in last chapter. The goals for Quick Builder are predefined goals from Ericsson, based on the requirements from section above. These are partly assembled from the requirements and partly as a result from competition in the multimedia market. These goals are:

Good usability: This goal involves the above requirements. When this goal achieves, the requirements achieve as well.

Drag-and-Drop functionality: This feature is heavily used in Web 2.0 applications, which open up for greater interactivity in the web application. It lets the user to drag an object, such as images or texts, from one certain area to another and finally drop it there.

Most used functions: As mentioned a couple of times before, frequently used functions. These could be any possible functions, but the target audience determines which ones are relevant.
Salesmen friendly: Salesmen do a lot of demonstrations for marketing the application and therefore it is important to know the application they are marketing. For this purpose it is appreciated if the application is easy to learn and preferably in a short amount of time.

No customization: Means that the application does not have to be adapted for each distinct customer’s circumstances, which has the positive effect of undemanding and less time-consuming setup.

2.3 Creation of high-fidelity prototype

2.3.1 The programming language

For developing the prototype a development framework called Google Web Toolkit (GWT) was used. GWT compiles written Java source code into optimized JavaScript code, which makes it possible to create AJAX-like applications without knowing any JavaScript. GWT can be used as a plug-in for Eclipse and get support for Java-like debugging. Since it generates optimized JavaScript code it has cross-browser support, resulting the web application looks the same across all major browsers. GWT also has good client-server communication support, allowing GWT’s simple RPC calls. The framework is open source and licensed under Apache License v2.0.

There is another API library, called GWT Ext, which also has been used in the development process. This library has many complete ready-to-use widgets, such as calendars and portals. GWT Ext is licensed under LGPL v3.0.

The GWT was chosen because of the features mentioned above. The goal of achieving good usability is almost equivalent with AJAX-like applications; under the condition it motivates the use of a certain feature. Java and the development environment Eclipse are language and program that I master. In addition, the client-server communication support with simple RPC calls was very welcomed when implementing the file upload function. Also the ready-to-use widgets, like the calendar widget, saved a lot of development time.

2.3.2 Implementation

The first objective was to create a high-fidelity prototype of Quick Builder. Figure 2.1 on next page shows the initial paper sketch supplied by Ericsson. The sketch can be considered as a low-fidelity prototype; since it is made of paper, has no functionality and does not support interactivity.
Based on the sketch above together with the goals from last section, the high-fidelity prototype was developed, see Figure 2.2.
Their appearances look almost the same, but differ a lot under the skin. The high-fidelity prototype supports interactivity for almost every feature, using both vertical and horizontal prototyping (both terms are described in Theory chapter).

Below follows features explanation of high-fidelity prototype in Figure 2.2.

**Portal and portlets**

The graphical user-interface is divided into five windows, called portlets. All these portlets lies in one big frame, called portal. Note that, this portal is a specific user-interface component (as well as the portlets) and should not be mixed up with the mobile portal the user intends to build with Quick Builder.

Each portlet is movable and can be drag-n-dropped anywhere in between the other portlets. Dropping a portlet between two others will make them move apart on behalf of the dropped one. In addition, each of them can be collapsed and carry small tools on the portlet header.

These portlets behaviors were not required in the initial plan. It was an intentional choice to bring along this feature, as it allows the user to arrange the portlets as the user like and hopefully improves the work efficiency.

The five portlets are titled Tools, “Options”, Workspace, Pages and Templates. The second one can be hard to find since it is not called Options. The reason is that the header carries various titles depending on the selected tool in Tools. In the case of Figure 2.2 the Options portlet is called Graphics. Thus, pressing the 11 different tools in Tools causes the Options portlet to display the respective tool, as well as the tool title in the header.
Pages and Templates
Pages portlet administrates all pages on the mobile portal. It is divided into three tabs; Thumbnails shows miniatures of available pages, Tree Structure shows all files and folders and Archive is another tree structure of files and folders that have been stored separately, for example from a campaign that may returns year after year. The Pages header has common tools for open, new, delete and copy pages.

Templates portlet allows the user to choose among several templates with different styles, layout and themes.

Both are not fully developed and has limited or none interaction.

Workspace
Workspace portlet has two tabs, Preview and Source. Both embody all content on the current page selected in Pages. The Preview displays how the page currently looks like whilst Source shows the corresponding source code. The header contains Undo, Redo, Save and Preview in device, where the last tool sends all Workspace content to a temporarily web address on the Quick Builder server when button is pressed. Then, the address pops up on the screen and is accessible from any mobile device.

Little consideration has been taken to adapt the Workspace content to a mobile phone. The first is the width of the Workspace that matches the screen width (240 pixels) of my working phone, to avoid horizontal scrolling. The second is to solely handle mobile phone-friendly content within the Workspace area that later can be generated to readable XHTML code when the content is going to be previewed in device. HTML elements like tables are totally omitted, since they are known to perform different in different devices.

Tools and Options
Tools contain 11 tool icons titled from left to right: Graphics, Text, Color, Services, Menu, Form, Themes, Campaigns, Reports, Publish and Settings. The ambition is not to give an in-depth description of each tool, but a short one:

Graphics: Lets the user upload images to Quick Builder for putting them on Workspace. All uploaded images are kept in a container, called Store. From here images can be drag-n-dropped on Workspace.

Text: Gives the user a traditional text editors with most used functions, such as bold, italic, underlined, font, size, colors, hyperlink, bullet list and text alignment. The text can easily be added to Workspace by pressing the add button in the bottom. By click on any text in Workspace makes Options portlet switch to Text mode and the text will appear in the text editor, ready to be edited. In text edit mode, the same text appears in both editor and Workspace. That means, a single change in text editor, the same change on Workspace, all happen in real-time.

Color: Lets the user change the color of text on Workspace. Note that Color tool was meant to change more than just the text color, like Workspace background, certain areas and borders.

Services: This tool contains a library of third part services and allows the user to put them on Workspace. The tool was not fully developed with limited interaction.

Menu: This tool contains different menu styles, with support for some customization. The user can put them on Workspace. The tool was not fully developed with limited interaction.

Form: This tool contains the most common HTML forms, such as input fields, radio buttons, check boxes, buttons and select lists. The tool was not fully developed with limited interaction.
**Themes:** This tool contains a list with applicable themes. The tool was not fully developed with limited interaction.

**Campaigns:** This tool contains a calendar and input fields for planning campaigns. For example, the user can plan a Christmas campaign and set certain pages or banners to be available only in month of December. The tool was not fully developed with limited interaction.

**Reports:** This tool shows click statistics for each page in the mobile portal, visualized in Flash based bar graphs. The tool was not fully developed with limited interaction.

**Publish:** This tool contains, side by side, two tree-structures of folders where files can be transferred from the private computer to the online server of Quick Builder. Think of FTP client software. The tool was not fully developed with limited interaction.

**Settings:** This tool contains general settings for device and server. The tool was not fully developed with limited interaction.

**Right click**
Right click on Workspace is supported. The right-click menu contains: Set visibility, Copy, Cut, Paste, Delete and Properties. The Set visibility lets the user set rules for certain target groups. Features like Copy, Cut, Delete and Properties work on elements in Workspace, whereas Delete is the only one of these that actually works.

Before the development of high-fidelity prototype begun, a priority list was made over features that needed to be developed:

- **Tools:** Graphics, Text, Color, Services, Menu, Form, Campaign, Reports and Publish
- **Workspace:** Preview mode and Source
- Pages and Templates support
- Undo, Redo and Save
- Preview in device

Many of these appear in the sketch, see Figure 2.1. The plan was the have some functionality and interactivity for respective feature in the list above.

During the implementation phase new features and functionalities were discovered, as a result of discussion with the supervisor at Ericsson. As the thesis was closing in on the final deadline, the priority list of features was modified. Some features became more important to finish than others, but still all tools had to be present and contain any content, no matter if it was dummy content or not. The most important features were Graphics, Text and Color together with the Preview in device and Preview mode in Workspace. These were vertically prototyped, and also Campaign and Reports in some sense. The rest were horizontal prototyped.

How well the goals were achieved is shown later in Conclusion.
2.4 Creation of low-fidelity prototype

The low-fidelity prototype was created as a copy from the high-fidelity prototype. It was made of OH papers (transparent plastic film) on which the graphical user interface were drawn in different colors. The ambition was to match the high-fidelity as much as possible. Few shapes of icons and images where hard to imitate due to the thickness of the pen used, just resulted in unidentifiable blotches. The solution was to compromise the look by drawing a slightly different icon but with the same conceptual meaning. It is important to have these two fidelities identical in terms of appearance and functionality for following comparison with each other. Figure 2.3 shows the low-fidelity prototype.

The high-fidelity prototype allows good interaction. For example, if the user mouse clicks on the text tool the application responses by showing text options in a separate window beneath the toolbar. This kind of window support or paging is also simulated in the low-fidelity prototype. Each window, text, image or other element are made on a separate piece of OH paper. Thanks to their transparency they can easily be piled and still visible through all the layers. That was the primary reason the OH paper were chosen over the ordinary paper.

Hereafter, the prototype in Figure 2.3 is the one referred when mentioning the low-fidelity prototype in the rest of the thesis, not the initial sketch in Figure 2.1.
3. Theory

This chapter is focusing on the theory of prototyping, statistical analysis and related work.

3.1 Prototyping

A prototype makes it possible to gather stakeholders and make them share the same vision. (Preece et al., 2002)

A prototype can be anything from a piece of paper to advanced software. The purpose of making a prototype is to save time and money. That can be done by lowering the number of features compared to the full-functional product. Each reduction has its own application area. Building a prototype with a small number of features is called vertical prototyping. The few features are compensated by having a deep functionality, which makes it possible to carry out realistic user tests on these. The second way of compromising is horizontal prototyping, generally including a full-featured user interface with little or almost none functionality. Even though this prototype has little functionality it is still good enough for letting users test the whole user interface. In addition the prototype can be implemented quickly. (Nielsen, 1993)

It could be tempting to reuse parts of the prototype into the final product since the developers probably offered some time developing the prototype, but is it wise? There are two approaches when developing a prototype, evolutionary prototyping and throw-away prototyping. Evolutionary prototyping means that the prototype evolves into the final product. For evolving the prototype into the final product several tests are required during the process, ensuring the prototype still fits the user requirements. The decision of whether do this approach or not, cannot be postponed to the future. The approach needs to be planned from the very beginning; otherwise the prototype is not going to end up as a robust product. The other way is throw-away prototyping. It simply means that the prototype is thrown away after its purpose is achieved, taking the knowledge from the prototype and start building the product from scratch. (Preece et al., 2002)

3.1.1 Different fidelities of prototype

There are two different fidelities of prototypes covered in this thesis, low- and high-fidelity. A low-fidelity prototype is a basic model of the final product. The material used in this kind of prototype is not the same as the material used for the final product. It does not look much like the final product either. The low-fidelity prototypes usually include paper-based or cardboard representations, such as outline, storyboard and index cards. It is often recommend using low-fidelity prototypes in the beginning of the development process, since it is a rapid way of prototyping. Thanks to its rapidness it communicates visions instantly to stakeholders (Preece et al., 2002). In addition, a paper-based prototype is easily editable and elicits one to make suggestions. Simple sketches like these also help a develop team to focus their discussion on higher level of issues (Wong, 1992).

A high-fidelity prototype is, in contrast to low-fidelity, very much like the final product both in look and use of material. Normally it is both fully functional and interactive.

The characteristics are many for both fidelities, next page shows a list their advantages and disadvantages, compiled and taken from the table of “Relative effectiveness of low- vs. high-fidelity prototypes” (Rudd et al., 1992).
Low-fidelity prototype

*Advantages:*

- Lower development cost
- Evaluate multiple design concepts
- Useful communication device
- Address screen layout issues
- Useful for identifying market requirements
- Proof-of-concepts

*Disadvantages:*

- Limited error checking
- Poor detailed specification to code to
- Facilitator-driven
- Limited utility after requirements establishment
- Limited usefulness for usability tests
- Navigation and flow limitations

High-fidelity prototype

*Advantages:*

- Complete functionality
- Fully interactive
- User-driven
- Clearly defines navigational scheme
- Use for exploration and test
- Look and feel of final product
- Serves as a living specification
- Marketing and sales tool

*Disadvantages:*

- More expensive to develop
- Time-consuming to create
- Inefficient for proof-of-concept design
- Not effective for requirements gathering

Besides the low- and high-fidelity, Yasar (2007) means there is a third fidelity type. It is mixed-fidelity prototype, which is a combination of the first two. A mixed-fidelity prototype shares the advantages from both fidelities. Some areas where high-fidelity is not necessary can be replaced with a fast paper-based low-fidelity prototype. Thus, it makes it more cost effective than a complete high-fidelity prototype. Furthermore, he exemplifies a web-based prototype of mixed-fidelity. The prototype can have small functionality and visual refinement of low-fidelity, but high-fidelity data model and level of interactivity.
3.2 Statistical analysis

3.2.1 Hypothesis testing

A hypothesis is a statement of what is expected to happen. There are two opposite hypothesis $H_0$ and $H_A$ to be considered in hypothesis testing. The latter defines the alternative hypothesis, which usually is the expectation supported. The other one defines the null hypothesis, which occurs when the alternative hypothesis is not true. Testing a hypothesis can be done in two ways, either by state that:

“$A$ differs from $B$” vs. “$A$ does not differ from $B$”, or

“$A$ is less than $B$” vs. “$A$ is more than $B$”

The first statement does not consider any direction, meaning that $A$ can be either more or less or even equal to $B$ and still meet the statement of “$A$ differs from $B$”, this is called two-tailed hypothesis. The second statement (“$A$ is less than $B$”) shows a single direction and is called one-tailed hypothesis. The two-tailed hypothesis is used in this thesis, since the stated question is to find out if there is any difference between the two fidelities in the context of usability testing.

The stated hypotheses are:

$H_0$: There is no statistical difference between the mean errors the two groups made while testing the low- and high-fidelity prototype.

$H_A$: There is a statistical difference between the mean errors the two groups made while testing the low- and high-fidelity prototype.

If the stated prediction $H_A$ is correct, then the null hypothesis will be rejected and vice versa. (Trochim, 2009)

3.2.2 Welch’s $t$-test

A $t$-test determines if there is a statistical difference between means from two groups relative to the spread of variability of their scores. It is appropriate to use in conjunction with hypothesis testing and commonly used for scores that follows a normal distribution. The $t$-test formula describes a ratio with difference between group means as numerator and variability of groups as denominator.

$$t = \frac{\text{difference between group means}}{\text{variability of groups}}$$

(Trochim, 2009)

The numerator describes a difference between group means ($\bar{X}$). Each mean is calculated as following:

$$\bar{X}_1 = \sum_{i=1}^{n_1} \frac{X_{1i}}{n_1}$$

$$\bar{X}_2 = \sum_{i=1}^{n_2} \frac{X_{2i}}{n_2}$$
\( X_{1i} \) is independent and \( N(\mu_1, \sigma) \) and \( X_{2i} \) is independent and \( N(\mu_2, \sigma) \). Each describes the estimated score. \( n \) is sample size for each group.

The both sample groups intend to have unequal variances since different fidelities are tested, and thus turning the Student’s \( t \)-test to a Welch’s \( t \)-test. The variances are:

\[
\begin{align*}
\text{Var}_1 &= \frac{1}{n_1-1} \sum_{i=1}^{n_1} (X_{1i} - \overline{X}_1)^2 \\
\text{Var}_2 &= \frac{1}{n_2-1} \sum_{i=1}^{n_2} (X_{2i} - \overline{X}_2)^2 
\end{align*}
\]

\( \text{Var}_1 \) and \( \text{Var}_2 \) are the variances for the respective groups.

(Råde & Westergren, 1998, p. 490)

The Welch’s \( t \)-test formula gives:

\[
t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{\text{Var}_1}{n_1} + \frac{\text{Var}_2}{n_2}}}
\]

When using this in significance testing the degree of freedom is calculated as:

\[
D.F. = \frac{\left( \frac{\text{Var}_1}{n_1} + \frac{\text{Var}_2}{n_2} \right)^2}{\left( \frac{\text{Var}_1}{n_1} \right)^3 \left( \frac{n_1 - 1}{n_1} \right) + \left( \frac{\text{Var}_2}{n_2} \right)^3 \left( \frac{n_2 - 1}{n_2} \right)}
\]

The value of \( t \) and D.F. are then used with a \( t \)-distribution table to test the null hypothesis whether two groups of means are equal.

(Student’s \( t \)-test – Wikipedia, 2009)

A two-tailed hypothesis test is used for that, which was introduced in last chapter. Finally a \( p \)-value is given that shows whether to reject the null hypothesis or not.

(Welch’s \( t \)-test – Wikipedia, 2009)

### 3.3 Related work

The intention of including this related work is to show similarities and differences to this thesis work. There are several works in the area of studying the difference of prototype’s fidelities, but there is specially one work that is highly related to this thesis. It is a paper “Usability Problem Identification Using Both Low- and High-Fidelity Prototypes”, written by Robert A. Virzi together with Jeffery L. Sokolov and Demetrios Karis. In this paper they are describing two experiments where the intention is to answer to the following question:

“In the later stages of user-interface design, are low-fidelity prototypes as effective as high-fidelity prototypes in identifying usability problems?” (Virzi et al., 1996, p. 236).
Two experiments were conducted for comparing usability problems between low- and high-fidelity prototypes.

The first experiment was a portable electronic-book that actually was bought in store; it was a final commercial product. The low-fidelity prototype was created with a simulation of the screen and keyboard on a paper. Pictures and text used for the prototype were based on the screens from the real product. The user interacted with the prototype by pressing any button on the paper keyboard while thinking aloud. As the user pressed a button, the test leader simulated the action as what would happen in the real product and by changing the picture.

20 students were recruited and were paid a small fee for participating as test users. Every user was given the same three tasks to test in the prototype. Half the users were testing the low-fidelity prototype whilst the other half was testing the high-fidelity prototype. Every test session lasted for 35-40 minutes and was video captured.

The result showed that a total of 38 distinct usability problems were indentified over both fidelities. The proportion between the two groups was analyzed. They tried to find out if any fidelity was more sensitive than the other in identifying usability problems. The average user of high-fidelity group found 38% of the problems and the average user of low-fidelity group found 34% of the problems. The two fidelity groups did not differ significantly in their overall sensitivity. Finally, they studied the fidelities for shared problems. Of 38 total usability problems, were 34 found by the high-fidelity group and 32 found by the low-fidelity group.

In the second experiment they used a interactive voice response system for telecommunication service. For interacting with the high-fidelity prototype, users were calling the computer from a touch-tone telephone. For interacting with the low-fidelity prototype, users were calling one of the experimenters, who acted the role as the computer. Users and experimenter were sitting in the same room. On requests from the user, the experimenter pressed the buttons on the unplugged telephone while reading out loud what the computer would have said.

As in the first experiment, 20 students participated and were divided into two equal sized groups. One group tested the low-fidelity prototype while the other tested the high-fidelity prototype. Users were asked to think aloud and tests were video captured.

The result showed that a total of 21 distinct usability problems were indentified over both fidelities. The average user of high-fidelity group found 40% of the problems while the average user of low-fidelity group found 46% of the problems. Of 21 total usability problems, were 19 found by the high-fidelity group and 20 found by the low-fidelity group. It shows that both fidelity groups discovered almost the same problems.

They also stressed that high-fidelity prototype found some problems that the low-fidelity prototype did not find and vice versa. Besides, this does not necessarily mean there is a substantial difference; it could likely depend on the normal variability.

(Virzi et al., 1996)
4. Methodology

This chapter gives a short overview of the methods used for collecting information. Each method is described. Finally, it is declared how reliability and validity have been handled.

4.1 Structure of study

Two different types of studies and one prototype evaluation were done.

At first a literature study was made in the field of prototypes. That gave general knowledge about prototypes which later were narrowed down to software prototyping since it is more relevant to this thesis.

The second was a technology study wherein the aim was to learn about which technique to use for creating such an application. The numbers of appropriate technologies were delimited by the employer since they’ve already done internal research. Among this selection, the best technology for this purpose was picked out. Other technologies were used but in less extent.

For the prototype evaluation five usability tests accompanied with post questionnaires were put through.

4.2 Literature and technology search

The literature search was mainly done via the Internet since there are many searchable databases available. The most frequently used databases were the library of KTH and ACM Digital Library. The latter contains many proceedings, newsletters and magazines in the area of Human-Computer Interaction. A few sources were also found via other book’s literature recommendations.

The following search words and combinations of them were used:

- Prototype/Prototyping
- High fidelity/Low fidelity
- Evolutionary prototyping
- Throw-away prototyping
- Evaluation
- Software Development

Prototyping and High/Low fidelity are the most fundamental keywords for this thesis and were found in both books and Internet databases.

The technology search basically contained of fresh and updated information about JAVA programming language from the Internet. Normally there are lots of books about programming, especially about JAVA, but this framework is new and comes with some new program libraries for graphical interface. The framework is called Google Web Toolkit and information about it is mostly collected from their site. (Google Web Toolkit – Google Code, 2008)
4.3 Usability test

In a usability test, users within the specified target audience are observed by people from the development team, while testing a product by performing certain tasks. The product can be a website, application, or other. The test can be documented by video and audio capture as well as taken notes. Depending on the thoroughness of the test, a set of data can be gathered from the test, like errors and time for each task. A number of measurable usability goals can be set before conducting the test for future comparison purposes. The purpose of using usability tests is to identify usability problems or validate the concept of the product. The sooner problems are identified and fixed, the more it saves money and time.

The process iterates, beginning with prototype development, conducting user tests, analyzing the results and then fixing the usability issues. This cycle continues until the development team is satisfied. (Learn About Usability Testing-Test and Refine | Usability.gov, 2009)

One of the questions in this thesis was to find out what importance does a prototype’s level of fidelity have for the usability test. The first step was to prove the assumption of the fidelity’s importance in usability testing. As mentioned in the introduction, two versions of Quick Builder with different fidelities were made and evaluated. The tests for low-fidelity prototypes were conducted within the frames of this thesis and the high-fidelity within the collaborator’s thesis.

The ambition of usability testing is to receive valuable feedback from real users by letting them test an application. Meanwhile, data about how the user interacts with the application can be gathered and later used to identify usability and design issues. Based on this it is very applicable method for evaluating Quick Builder.

To get a richer quality feedback from the user than just observing what the person is doing, the user was asked to think aloud while interacting. According to Nielsen (1993) it generates quality data from a small number of users. The experimenter not only sees what the user is doing. With this method, user’s thoughts are spoken out, which reveals why the user acts in a certain way. However, it also means that the method has several disadvantages. Not only that it can be perceived as unnatural, it can also slow down the user and thus impact the test result.

From the usability test, the amount of errors and time are counted for each user and assignment. The definition of one error is either when a user clicks on the wrong spot, enters the wrong menu or reads the wrong value. As a result the error is subcategorized into wrong menu choice and miscellaneous errors. In comparison between the both fidelities, the error categories were merged into one category of total errors. To decide whether an error is an error can be tricky. It has to be in relation to what the user intends to do according to the assignment. The intention easily reveals when the user thinks aloud. Be aware of that an error was never judged by the user’s thoughts, only by the user’s actions.

4.3.1 Test user

The audience group for Quick Builder contains people with professions like journalist or administrator. Their average computer skill is somewhat limited. For that reason test users from the audience group were preferable. However, it was very difficult to find test users and no one was allowed to contact and use any of Ericsson’s customers, but the companies own customer contact experts. Otherwise Ericsson’s customers would be desirable since they are a part of the audience group. The search for users moved on and finally five users were found within my circle of acquaintances. Neither the number of users nor the social connections to me are optimal for user tests. The number of users is too small to draw any statistical differences of test results between low- and high fidelity prototypes. Furthermore, the users may be biased due to the connection to me. Unfortunately, this is the solution you get without any budget.
Note that testing five users is small enough to detect a satisfying amount of usability problems. (Nielsen, 2000)

Besides, this selection of users needed to be different from the collaborator’s selection in order to avoid impact from the first prototype tested to the second one tested.

The test users for low-fidelity prototype are described in the following list:

- **Test user 1**: Journalist, female around 30 years old. Modest computer skill without any earlier experience in programming or web design.
- **Test user 2**: Controller, male 49 years old. Modest computer skill without any earlier experience in programming or web design.
- **Test user 3**: Student of Media technology, male 23 years old. Very good computer skill with experience in both programming and web design.
- **Test user 4**: Student of Human resources, female 22 years old. Modest computer skill without any earlier experience in programming or web design.
- **Test user 5**: Student of Systems science, female 22 years old. Modest computer skill with little experience in programming but no web design.

The requirement level for participating in the test was at least to have modest computer skill and feeling comfortable using word processing programs and suchlike.

4.3.2 **Test environment and equipment**

The tests were usually conducted in the user’s home, in a quiet and undistracted room where the user could feel comfortable. Cell phones and other communication channels that were not a part of the test were turned off. The test took place by a table with one user and one experimenter, facing each other. On the table, between the two persons, the low-fidelity prototype was positioned. A laptop with in-built webcam was recording the interaction on the table and coincidentally recording the sound via an external headset microphone pointed at the user. For typing purpose the user was equipped with four colored pens and a piece of OH paper, instead of a keyboard. The user’s hand was used as the mouse cursor.

4.3.3 **Description of test procedure**

All tests follow the same pattern and therefore a typical test is described as follows. Before the test started the user were asked to read an introduction text about the test’s purpose, goal and procedure (see Appendix A: Usability test). Next, the user carefully read each question and performed the belonging assignment. As mentioned before the user was asked to think aloud. Then the user became silent, the person were cautiously encouraged to take up the think aloud methodology. Occasionally the user tried to ask leading questions or about what happens if a certain button is pressed. Neither question was answered due to the test rules. Although a few humph and sighs might have been uttered by the experimenter, subconsciously and of course not on purpose. Unlike the high-fidelity prototype, the low-fidelity does not have natural support for titled icons. By hovering the cursor over an icon in high-fidelity, a balloon tip appears that shows the icon’s title. To equalize low-fidelity with high-fidelity prototype, questions about the icon’s title were willingly answered. Questions about the test were also acceptable.

In addition to answer questions and conduct the test, the experimenter also has the role as a computer. Every press that required an action, such as a window changed content or an element was moved somewhere else in the application, was physically performed by the experimenter by moving OH layers around. The intention of this is like earlier, to equalize the interaction procedure of low-fidelity with high-fidelity prototype between user and computer.
4.4 Questionnaire

Questionnaire is a method for collecting data about users’ opinions within a certain group. The questions can be closed, gives the user a fixed number of answer options, or open, lets the user answers freely. The method could be used in conjunction with other methods, such as usability testing, and thus providing deeper understanding. Preece et al. (2002) advise to make questions preferably closed, clear and unambiguous. They also mean that alternating the order of a negative and positive questions, such as “Do you think this design looks bad?” or “Do you think this design looks good?” prevents the user from rushing through all questions without reflecting over the questions. Likert scale is used when users’ opinions need to be measurable. For example it can measure user satisfaction with a scale of strongly disagree, disagree, agree and strongly agree. (Preece et al., 2002)

The intention of this questionnaire was to gather data that shows what the users felt about the prototype. For instance, a user that never learns how to find the text tool may experience frustration and if the users think the prototype looks good they might want to express that. The purpose of the questionnaire is to gather this kind of emotions and also make it emotional comparable to the high-fidelity prototype.

The questionnaire consists of 10 questions that are related to the test of Quick Builder and was answered right after the test. All questions are closed-ended, except for the last one that is both closed-ended and open-ended. An open-ended question is a question that allows the respondent to formulate an own answer. Closed-ended question is the contrary to the open-ended, a traditional question with a few predefined answers for the user to pick.

The answers are based on the Likert scale, with a few answers that distinguish in level of agreement. The user picks the level of agreement that match the answer best. This is not an ordinary questionnaire. What makes this special is the answer scale, where the middle option (neither agrees nor disagrees) is removed. The removal causes the user to choose either side of agreement, a so called forced choice. (Ipsative - Wikipedia, 2009)

The questionnaire can be found in the Result chapter.

4.5 Reliability and validity

4.5.1 Reliability

Reliability is a measure of what level an instrument or approach provides equally results in two different situations under identical circumstances.

(Bell, 1993)

The reliability has been ensured in four different perspectives. These are the environment, user test, post questionnaire and the report itself.

To achieve great reliability the test took place in a neutral, quiet and relaxed room with low risk of interruption.

A booklet has been made consisting of thoroughly described introduction and test tasks. The written test tasks do not change from time to time, which give all users the same basic condition to follow and perform the same task.
Each question in the post questionnaire is carefully designed, in agreement with the supervisors, to decrease the ambiguity. A bad formulated question can cause diverse interpretations for different informants. One pilot test has been put through for reconsideration of questions, purpose of the questions, background story and approach.

Beside of conveying the content of this thesis there has been a strong focus on objectivity and accurate information gathering and performance. Thus the report should contain minimum of biasing.

By considering these perspectives the reliability has been strengthen.

4.5.2 Validity
Validity is a term that describes how much a thing measures what it is expected to measure. This has been ensured by making pilot tests where both questions and task list has been put under pressure. Right after the test and questionnaire there has been a discussion about the test which led to refinement of both test approach and some questionnaire questions. This feedback together with literature study and cooperation with the supervisor at Ericsson has made them representative. To avoid ambiguity in questionnaire answers, the middle alternative has been removed forcing the informant to make a side decision.

The usability tests intend to measure amounts of error and time for each user and assignment. These parameters can be extracted from recordings of both video and audio for comparison with high-fidelity test results.

The questionnaires assemble measurable user impressions for later comparison with high-fidelity questionnaire results.
5. Results

This chapter shows results of from both user tests and questionnaires. Each result section contains results from the low- and high-fidelity prototype.

5.1 Results from user tests

All user test material was originally written and performed in Swedish, but has been translated into English to make the language of the thesis uniform.

The methodology chapter described that the purpose of the user tests are to identify usability and design issues in Quick Builder and get some valuable feedback from real users within the target group. The results from user tests are divided into low- and high-fidelity sections. The methodology chapter showed that the user was given a booklet of test tasks to perform. Every test has generated qualitative data. In order to compare data from the various user tests of low- and high-fidelity prototypes, two tables of data have been extracted for this purpose. Note that number of errors includes both wrong menu choices and miscellaneous errors.

Table 1 on next page shows the number of errors five users made in each task during the user tests of low-fidelity prototype. This table is extracted from the transliteration from Appendix B: Test results.

Table 2 shows the number of errors four users made in each task during the user tests of high-fidelity prototype. This table is extracted from the collaborator’s transliteration.

Explanation of Table 1-2:

- These task descriptions are shortened versions of the relatively longer role-based questions from the original test tasks (see appendix A: Usability test).
- These tasks are numbered from 1 to 10.
- Each capital letter in the tables represents one unique user and test session.
- The numbers located beneath the capital letters are numbers of errors the user made in each task.
Table 1: The number of errors user made in user tests of low-fidelity prototype

<table>
<thead>
<tr>
<th>Nr</th>
<th>Short task description</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Mean error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Find the number of unique visitors.</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1,8</td>
</tr>
<tr>
<td>2</td>
<td>Plan a specific campaign.</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0,2</td>
</tr>
<tr>
<td>3</td>
<td>Create a bulleted text list, change fonts and add it to workspace.</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Add an image to workspace.</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1,2</td>
</tr>
<tr>
<td>5</td>
<td>Change text color of each row in the bulleted list.</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Let the image and text list switch places in the workspace.</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0,8</td>
</tr>
<tr>
<td>7</td>
<td>Replace the old image in workspace with a new image by first deleting the old one.</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>3,6</td>
</tr>
<tr>
<td>8</td>
<td>Preview the page in a cell phone.</td>
<td>0</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>Find the settings for customized content.</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>2,6</td>
</tr>
<tr>
<td>10</td>
<td>Build a page using a given outline.</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4,2</td>
</tr>
</tbody>
</table>

Table 2: The number of errors user made in user tests of high-fidelity prototype

<table>
<thead>
<tr>
<th>Nr</th>
<th>Short task description</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>Mean error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Find the number of unique visitors.</td>
<td>3</td>
<td>21</td>
<td>20</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Plan a specific campaign.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Create a bulleted text list, change fonts and add it to workspace.</td>
<td>1</td>
<td>25</td>
<td>4</td>
<td>0</td>
<td>7,5</td>
</tr>
<tr>
<td>4</td>
<td>Add an image to workspace.</td>
<td>3</td>
<td>6</td>
<td>11</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Change text color of each row in the bulleted list.</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0,25</td>
</tr>
<tr>
<td>6</td>
<td>Let the image and text list switch places in the workspace.</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0,25</td>
</tr>
<tr>
<td>7</td>
<td>Replace the old image in workspace with a new image by first deleting the old one.</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Preview the page in a cell phone.</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>1</td>
<td>8,25</td>
</tr>
<tr>
<td>9</td>
<td>Find the settings for customized content.</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1,25</td>
</tr>
<tr>
<td>10</td>
<td>Build a page using a given outline.</td>
<td>5</td>
<td>21</td>
<td>15</td>
<td>12</td>
<td>13,25</td>
</tr>
</tbody>
</table>
5.2 Results of shared usability errors

There are several equal errors discovered in both fidelities. These errors are called shared errors.

Table 3 shows what error that was discovered by both fidelity groups. First column is the assignment 1-10 and next column is the common error. Some errors were discovered by more than one user within a fidelity group, it is shown by the number of users in respective fidelity.

Table 3: The number of shared errors user made for respective fidelity of prototype

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Error</th>
<th>Low-fidelity [number of users]</th>
<th>High-fidelity [number of users]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>• Clicking on wrong tab in <em>Reports</em></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>• Searching for unique visitors in <em>Graphics</em></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>• Entering <em>Menu</em></td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• Entering <em>Form</em></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>• Searching for graphics in <em>Graphics</em></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• Adding graphic to <em>Workspace</em></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>• Deleting a graphic on <em>Workspace</em> by clicking on the red cross in <em>Store</em></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>• Deleting by pressing the delete button</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>• Entering <em>Publish</em></td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>• Mistaking <em>Properties</em> for being <em>Set visibility</em></td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>• Creating a list instead of importing a ready service from <em>Services</em></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• Searching for how to add links, entering <em>Services</em></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• Searching for how to add links, entering <em>Menu</em></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>33</td>
<td>17</td>
</tr>
</tbody>
</table>

The total number of shared errors for both fidelities is 50. Note that all of these cannot be seen as usability problems. For example, if one user enters the Form tool twice, it will be counted as two errors but there is only one usability problem detected. This means, for this data of errors, it cannot be directly compared to Virzi’s et al. (1996) evaluation results, since they were counting usability problems. What can be done is to compare what kinds of problem the different prototypes discovered. This will later be picked up in the Conclusion chapter.
5.3 Results from questionnaires

The questionnaire was originally written and performed in Swedish, but has been translated into English to make the language of the thesis uniform.

As mentioned in the methodology chapter, the purpose of these questionnaires is to catch the user’s impressions from Quick Builder, which is qualitative data, and turn it into measureable data.

The two following tables on the next page show the questionnaire results from low-fidelity group, in Table 4, and high-fidelity group, in Table 5. Table 5 comes from the collaborator’s questionnaire results. The results can also be seen in Appendix C: Questionnaire, then in forms of comparison charts.

Explanation of Table 4-5:

- Statements are numbered from 1 to 10. Note that these numberings has nothing to do with the numberings of Table 1-2, they just happened to have the same number of digits.
- Each capital letter in the tables represents one unique user and test session. These are the same unique users as those from Table 1-2.
- The numbers located beneath the capital letters are ratings:
  - 1 = Strongly disagree
  - 2 = Disagree
  - 3 = Agree
  - 4 = Strongly agree
- Statement 10 is designed to receive both a rating and additional opinion in text form. See Appendix B: Test results for the textual opinion.
Table 4: Questionnaire results from low-fidelity prototype

<table>
<thead>
<tr>
<th>Nr</th>
<th>Statement</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Mean rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>It is difficult to get around in the application.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1,2</td>
</tr>
<tr>
<td>2.</td>
<td>I can quickly find what I’m looking for in this application</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>This application seems logically structured</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3,4</td>
</tr>
<tr>
<td>4.</td>
<td>This application would need more explanation, initially.</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5.</td>
<td>I think the application’s look is very appealing.</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3,4</td>
</tr>
<tr>
<td>6.</td>
<td>I feel like I’m in control when using the application</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2,6</td>
</tr>
<tr>
<td>7.</td>
<td>Learning how to locate things in this application is problematic</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>This application helps me to perform the tasks I want.</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3,2</td>
</tr>
<tr>
<td>9.</td>
<td>I feel frustration when I use this application.</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1,8</td>
</tr>
<tr>
<td>10.</td>
<td>I miss one/several features of this application (if so, what / which?)</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2,4</td>
</tr>
</tbody>
</table>

Text answer from statement 10:

A – A help section that describes approaches, such as insert picture, text, etc.
B – The delete button should remove elements.
C – A visible trash bin and a default view (a default view that is visible when no other tool is in use).
D – The delete button should remove elements.
E – The delete button should remove elements; alternatively you can remove an element by dragging and dropping it outside the workspace.
### Table 5: Questionnaire results from high-fidelity prototype

<table>
<thead>
<tr>
<th>Nr</th>
<th>Statement</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>Mean Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>It is difficult to get around in the application.</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1.75</td>
</tr>
<tr>
<td>2.</td>
<td>I can quickly find what I’m looking for in this application</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>3.</td>
<td>This application seems logically structured</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>This application would need more explanation, initially.</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2.75</td>
</tr>
<tr>
<td>5.</td>
<td>I think the application’s look is very appealing.</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>I feel like I’m in control when using the application</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3.25</td>
</tr>
<tr>
<td>7.</td>
<td>Learning how to locate things in this application is problematic</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>This application helps me to perform the tasks I want.</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3.75</td>
</tr>
<tr>
<td>9.</td>
<td>I feel frustration when I use this application.</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>10.</td>
<td>I miss one/several features of this application (if so, what / which?)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1.75</td>
</tr>
</tbody>
</table>

**Text answer from statement 10:**

- **F** – Nothing.
- **G** – Video?
- **H** – Nothing.
- **I** – Nothing.
6. Analysis

This chapter shows the statistical difference between the two fidelity results with the aid of Welch's t-test.

6.1 Comparison of results from the usability tests

The null hypothesis \((H_0)\) is:

There is not any statistical difference between the mean errors the two groups made when testing the low- and high-fidelity prototype.

This comparison makes use of the mean errors from Table 1 and 2 in previous chapter. Low-fidelity group is denoted \(x\) and high-fidelity group \(y\). Each task has a number \(i\) from 1 to 10. Every user is uniquely denoted from \(A\) to \(E\) in low-fidelity group and \(F\) to \(I\) in high-fidelity group. The number of users within a group is called sample size, denoted \(k\). Sample size of low-fidelity is 5 and high-fidelity is 4. The mean number of errors made in \(i\) task for the low-fidelity group is:

\[
x_i = \frac{\text{sum of errors the users made in task } i}{\text{sample size of users}} = \frac{(x_{iA} + \cdots + x_{iE})}{k_x}
\]

Likewise, the mean number of errors made in \(i\) task for the high-fidelity group is:

\[
y_i = \frac{\text{sum of errors the users made in task } i}{\text{sample size of users}} = \frac{(y_{iF} + \cdots + y_{iI})}{k_y}
\]

Furthermore, the sum of all mean errors in low-fidelity group is dived by the number of observations, \(n_x = 10\) and \(n_y = 10\):

\[
x = \frac{(x_1 + \cdots + x_{10})}{10} = \frac{1,8 + 0,2 + 2 + 1,2 + 1 + 0,8 + 3,6 + 4 + 2,6 + 4,2}{10} = 2,14
\]

The above is resulting in a table of mean errors for each task. Altogether they create a new mean error from the ten observations:

<table>
<thead>
<tr>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
<th>(x_5)</th>
<th>(x_6)</th>
<th>(x_7)</th>
<th>(x_8)</th>
<th>(x_9)</th>
<th>(x_{10})</th>
<th>(\bar{x})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,8</td>
<td>0,2</td>
<td>2</td>
<td>1,2</td>
<td>1</td>
<td>0,8</td>
<td>3,6</td>
<td>4</td>
<td>2,6</td>
<td>4,2</td>
<td>2,14</td>
</tr>
</tbody>
</table>

In the same way does the high-fidelity group \(y\):

<table>
<thead>
<tr>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(y_3)</th>
<th>(y_4)</th>
<th>(y_5)</th>
<th>(y_6)</th>
<th>(y_7)</th>
<th>(y_8)</th>
<th>(y_9)</th>
<th>(y_{10})</th>
<th>(\bar{y})</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>7,5</td>
<td>6</td>
<td>0,25</td>
<td>0,25</td>
<td>2</td>
<td>8,25</td>
<td>1,25</td>
<td>13,25</td>
<td>5,18</td>
</tr>
</tbody>
</table>
These values above are put into standard deviation formula:

$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n_x - 1}} \approx 1,41$$

High-fidelity group is resulting $s_y \approx 4,93$. The Welch’s $t$-test formula gives:

$$t = \frac{\bar{x} - \bar{y}}{s_{x-y}} = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}} = \frac{2,14 - 5,18}{\sqrt{\frac{1,41^2}{10} + \frac{4,93^2}{10}}} \approx -1,87$$

The negative value of $t$ means that $\bar{y}$ is greater than $\bar{x}$. If they swap places the result would be positive $1,87$. The degrees of freedom are 10. According to a table for $t$-distribution $t(10) = 2,23$ at alpha significance level of two-tailed $p = 0,05$. The difference between the calculated $t = 1,87$ and $t$-distribution $t(10) = 2,23$ is considered to be not quite statistically significant. This implies that $t$ of 1,87 cannot reject the $H_0$. The null hypothesis is accepted, there is no statistical difference between the two groups’ number of errors.

---

1 See table for $t$-distribution (Blom et al., 2005).
7. Conclusion

This chapter answers the stated questions of the thesis.

7.1 Importance of fidelity level

There is no statistical difference between the mean errors the two groups made while testing the low- and high-fidelity prototype.

The result consists of amount of errors made from each user in respective fidelity. The importance of the fidelity level in this thesis is determined by the comparison of the test results from both low- and high-fidelity prototypes. Since the sample size did not consist of more than five users it is not recommended to draw statistical conclusion about to what level the two fidelities differentiate. However, without quantifying the less statistic significant result, a small tendency can be assumed that there is no difference of the two fidelities.

This tendency is also supported by Virzi’s et al. (1996) conclusion, whose tests were made under similar circumstances, but with twice the sample sizes.

What is the importance of the prototype’s fidelity level, in the context of usability testing, under the condition where the low-fidelity is created as a copy from the high-fidelity prototype?

If the difference in detecting errors between the fidelities is assumed to be zero, the result is saying that fidelity does not matter in the context of usability testing under the condition where the low-fidelity is created as a copy from the high-fidelity prototype. This is not completely true. Even if these fidelities did not statistically found different amount of errors, does not mean they found identical errors. The two fidelities actually found different kinds of problems. Generally the high-fidelity prototype found problems of higher severity (though it is hard to claim since no such data has been extracted from the transliterations) than the low-fidelity prototype. More of that is discussed in next chapter 9.3 Discussion about the results.

7.2 Guidelines for creating an interactive multimedia prototype

What guidelines for developing an interactive multimedia prototype can be extracted from the implementation and evaluation of the Quick Builder prototypes?

7.2.1 About the guidelines

These are guidelines for software prototyping within the interactive multimedia area. These guidelines should not be seen as a description of how Quick Builder was built, but rather as a result from usability tests, studies and experience from the Quick Builder implementation. The evaluation results from the two prototypes pointed at some similarities, but also at certain differences.

7.2.2 Guidelines

The fundamental idea of a prototype is to evaluate whether a design or concept is good or not. In contrast to the final product, it meant to be cheap and quick to build. Thanks to its less complexity it is also easy to make changes with the aid of inputs from user tests, heuristic evaluations or other valuable feedbacks.
• **Benefits of prototyping**

The benefits of prototyping were mentioned in the beginning of this section; it is inexpensive and rapid way of testing a concept compared to a final product. If you have decided to create a prototype before beginning with the final product, you might wonder which way is the best way of doing it. You can choose between low- and high-fidelity prototyping.

The low-fidelity prototype is inexpensive to develop since it is fast and also because it is usually made of a low-fidelity material, like paper, OH paper or cardboard. Thanks to its rapidness it is not very difficult to revise as feedbacks are received. This revision and feedback let you iterate over several prototypes until you are satisfied with the result. Moreover, the world does not fall apart when critics are received for a low-fidelity; you just modify the prototype. It is worse for the high-fidelity where the developer has offered both time and soul, which can lead the developer to feel despair.

According to the thesis result, the most common usability problem found in low-fidelity was the difficulty in finding certain tools (see Appendix B: Test results). Probably because of the rough impression the low-fidelity does, lines are skew, colors are vivid and icons are hard to identify because they are too cluttered. For some users it caused confusion as they mistaken one tool for being another. Moreover, many users did not think of that they had a right-click option when they tested the low-fidelity prototype.

The advantage of high-fidelity is that it is easier to test the prototype since it runs by itself and does not involve any other than just the computer and the test user, compared to the low-fidelity. It also gives richer feedback in terms of detecting logical-, font-, color-, proportion problems and software bugs. A practical example is when the user was asked to type some text with any font in the text editor and the user wondered if the font already was selected or not (it was not marked or anything in the list of fonts). Or when the user was asked to upload an image onto the image store and add it to workspace. The image was not found because the image was placed on another page that currently was not showed. The solution was to step forward to that page where the image automatically was placed and then drag and drop it on workspace. The high-fidelity performed different from the low-fidelity as it found other types of usability problems, of higher severity.

It is recommended to use the low-fidelity prototyping in the beginning where a lot of changes usually are required. If the budget does not permit anything else but low-fidelity prototyping throughout the whole process, then the demands are high for conducting computer-like usability tests. In addition, the low-fidelity prototype must be free from clutches and unidentifiable objects. The best way of having that, is to create the prototype in an image processing program and then carry out user tests on that.

Both fidelities are still necessary in the development process. It is not a matter of which fidelity to choose, it is a matter of for how long the low-fidelity prototyping should last and how soon the high-fidelity prototyping should supplant. The high-fidelity prototype is recommended to use after obvious usability problems have been detected and fixed by low-fidelity evaluations. That is because high-fidelity prototype is, relative the low-fidelity, more costly in both time and money.

• **Reusing of code in the real application**

There are philosophies about whether to reuse the code in the final application or not. This is presented in the Theory chapter with evolutionary and throw-away prototyping. As the chapter argues, if you are planning to evolve the prototype into the final product, the decision has to be made before the prototyping process starts.
The prototyping plan in the case of Quick Builder was somehow unclear. The intention was to create a prototype that has a usability value, drag-n-drop functionality; and a programming technical value, because of the choice of GWT, which Ericsson has showed an interest in. These preparations are not enough for evolutionary prototyping, but still a good input and inspiration for further development. It is profitable to have the same developers for prototyping as for the real application production, since they know the code. If you are determined to evolve the prototype into the final product, make sure you make a thorough plan for this in the beginning and make room for scalability. If it does not quite succeed, you can at least use parts of the prototype.

- **Selecting the key functions**
  Selecting the right key functions for a prototype does not have one right answer, as it depends on the prototype’s purpose. The purpose of the high-fidelity prototype of Quick Builder was to demonstrate it at an exhibition, which in this case excluded the low-fidelity as an option. With the ambition to make a good demonstration as possible, the prototype must be outstanding. This ambition demands having a few features that are fully functional, or at least enough for showing its potential. If the remaining “background features” have underlying code or not, is not a concern. But their appearances are still important for not making the prototype look nearly empty. Demonstration purposes requires more vertical prototyping than horizontal prototyping, these terms are also mentioned in the Theory chapter.
7.3 Prototype of Quick Builder

The goals for the prototype of Quick Builder are similar to the goals for Quick Builder. The prototype should strive for behaving and looking like Quick Builder, but since it is a prototype with short of time, money and resources it should not be completely like Quick Builder regarding planned functionality, reliability, performance, appearance, etc. Thus, the initial priority list for the prototype was compiled with required features. This is the same initial priority list as in section 2.2 Goals for Quick Builder:

- **Tools**: Graphics, Text, Color, Services, Menu, Form, Campaign, Reports and Publish
- **Workspace**: Preview mode and Source
- Pages and Templates support
- Undo, Redo and Save
- Preview in device

The feature goals above were partly achieved. As mentioned before, Graphics, Text, Color and Workspace, Preview mode and Preview in device were vertical prototyped with great functionality. Graphics can be uploaded to Store and drag-n-dropped on Workspace. The same goes for the Text tool with multi formatting support. A functional paint bucket from Color tool changes the color of texts in Workspace. Preview in device works properly, it sends all Workspace content to a temporary web address, accessible from any device. The remaining tools are accessible, but have limited functionality.

New feature goals were discovered along the process and were also implemented:

- Movable elements (texts, images, links, services, etc.) within the Workspace
- Movable portlets within the portal window
- Support for right mouse click, a menu consists of Set visibility, Copy, Cut, Paste, Delete and Properties. Underlying functionality for Set visibility and Delete
  - Delete elements from Workspace
  - Set visibility, allows the customer to control the content, based on target groups (visible, but not functional)
- Settings for server and device (visible, but not functional)

7.4 User opinions in questionnaires

The intention of the questionnaire method was to get deeper information about user’s impressions of the two prototype fidelities. Seen to the user test results, questionnaire results were not surprising. The user groups’ opinions about the prototype they tested do not differ remarkably much. According to differences in Table 4 and Table 5 from 5.2 Results from questionnaire, the three most extreme mean differences were -0.75, -0.65 and 0.65. The questionnaire scale is graded from 1 to 4. The differences do not even reach one step on the scale. The three questions are:

- This application would need more explanation, initially.  
  (high-fidelity group has a 0.75 higher agreed level than low-fidelity group)
- I feel like I’m in control when using the application.  
  (high-fidelity group has a 0.65 higher agreed level than low-fidelity group)
- I miss one/several features of this application (if so, what / which?).  
  (low-fidelity group has a 0.65 higher agreed level than high-fidelity group)

Appendix C: Questionnaire shows all questions and answers from both fidelity groups in a comparison charts. Note that median values are compared.
8. Discussion

This chapter discusses the analysis, results, methodology and finally gives suggestions for further work.

8.1 Discussion about analysis

According to the vague conclusion from section 7.1 Importance of fidelity level, there is no statistical difference between amounts of problems found in low- and high-fidelity prototype, under the condition that low-fidelity was created from the high-fidelity.

According to the results from Table 1 and Table 2 (in section 5.1 Results from user tests), 107 problems were found in low-fidelity and 207 in high-fidelity. You may ask how it is possible that there is no statistical difference between the numbers of problems. The statistical analysis made in this thesis considers standard deviation, which is a method of estimating the dispersion over a quantitative variable. The hypothesis theory is based on Welch t-test, which determines whether means from two groups are statistically different from each other or not. There are three possible explanations why the difference is not statistically significant. By studying the t-test formula the possible reasons can be revealed:

\[
t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}}
\]

Given that the difference is too small is equivalent to a small t. A small t is achieved either by a small numerator (difference between group means) or a large denominator (variability of group). In the last step of the formula, a large denominator is achieved either by large variances (v) or small sample sizes (n) for each group.

Since the sample size was rather small it could be the reason, while the remaining two are quite large. It could also be the variance, see table of high-fidelity group in chapter 6.1 Comparison of results from the usability tests, the high-fidelity variance is large, due to the irregularity in group. Some users caused extreme values that contribute to the large variance.

8.2 Discussion about results

Different types of data

In the Theory chapter Virzi’s et al. (1996) work was paid attention. The ambition was to point at similarities and differences between their work and this thesis work. But pointing at similarities should be done cautiously, since there are two main differences between the two work’s scope and approach, it could be somewhat misleading to make direct comparison of the two.

The list shows some differences in results from this thesis’ tests and Virzi’s et al. experiments:

- This thesis resulted in, of 314 total errors, were 207 found by the high-fidelity group and 107 found by the low-fidelity group

- Virzi’s et al. (1996) first experiment resulted that, of 38 total distinct usability problems, were 34 found by the high-fidelity group and 32 found by the low-fidelity group. Their second experiment, of 21 total distinct usability problems, were 19 found by the high-fidelity group and 20 found by the low-fidelity group.
Firstly, no number usability problems have been counted in low- and high-fidelity prototype of Quick Builder, only the number of errors each user made. How much impact it does on the relationship between low- and high-fidelity of Quick Builder has not been investigated, but there is likely an impact.

Secondly, their experiments consider distinct usability problems while the low- and high-fidelity prototypes of Quick Builder consider just a total number of errors, meaning that some errors are counted more than once.

There is likely an impact, since it happened in both prototypes of Quick Builder for instance that users entered a tool option (which is an error in a certain assignment) several times for the same assignment.

**Lack of severity data**

In the conclusion it is claimed that the high-fidelity prototype discovered more problems of higher severity than the low-fidelity prototype, without showing any data that supports that claim. Transliterations from both fidelities’ usability tests lack the data of errors’ level of severity, since it was out of thesis scope back then. The reason why the severity of the problem is mentioned in this discussion is to stress that there were differences between the fidelities, although severities were not thoroughly measured.

As mentioned before in third paragraph of Benefits of prototyping from last chapter, the low-fidelity prototype found many errors connected to wrong menu choices. This also happened in the high-fidelity prototype. Other problems detected in low-fidelity prototype is low-fidelity specific problems, like users did not remember the right-click option and that the user-interface is too cluttered for getting an acceptable overview of the prototype. Since they are low-fidelity specific problems, they will never be a problem in high-fidelity prototype, and thus never a problem in the final product.

My collaborator’s thesis shows many of the errors appeared in his high-fidelity prototype tests. Few of them were also mentioned in Benefits of prototyping from last chapter, the fourth paragraph. These types of error are far more severe than those occurred in low-fidelity prototype.

**Possible impacts on the results**

It is obvious that the high-fidelity discovered more usability problems than the low-fidelity, 207 respective 107. There are several possible reasons why the gap is so big between the two fidelities. A few possibilities are listed below:

- The test user did not belong to the stated target audience for Quick Builder. That was the case for both fidelity sides though, due to difficulties in finding right users. Using test users from outside the target audience may perform extreme values, causing large variances and distortions of the mean errors.
- It is plausible that a user who tests a high-fidelity prototype feels more comfortable, less pressure and independent from test leader, compared to a low-fidelity test where the test leader is a part of the interaction between the user and the prototype. The comfort feeling could be a reason why the high-fidelity user is more exploratory than reductive, and thus makes it easier to run into errors.
- Another difference for the user in testing a low-fidelity compared to a high-fidelity prototype, is that the low-fidelity prototype does not return any feedback, such as showing when a button is pressed, drop-down-list does not disappear when the user click an option, etc. Whether the feedback is a helping or ruining the performance, it may have some impact in either direction.
Since the low-fidelity prototype is a copy of the high-fidelity prototype, it is likely the high-fidelity users discover nearly the same errors as the low-fidelity provided, plus additional errors that the rich details, interactions and bugs contribute. The errors found in low-fidelity prototype can be seen as a subset to the errors found in the high-fidelity prototype. It is likely that the high-fidelity prototype always finds more errors than the low-fidelity prototype, since it is a subset of the high-fidelity prototype.

To sum up, Virzi’s (1996) result mean that high-fidelity prototypes discover the same amount of problems in both fidelities. The condition is that their high fidelity prototype was a fully functional commercial product and the low-fidelity was created based on the high-fidelity prototype. This thesis result means that high-fidelity prototypes discover more problems of higher severity than low-fidelity prototypes do.

The difference is that this high-fidelity prototype does not have a final look and performance as a commercial product, like in the case of Virzi (1996). This may lead to greater error differences between the respective low- and high fidelity prototype.

**8.3 Discussion about methodology**

**Difficulty in conducting usability test on low-fidelity prototype**

As shown in Results, the low- and high-fidelity prototype resulted in 107 respective 207 errors, 50 of these are identical errors. Theoretically it is possible to discover almost the same problems in both fidelities, if the low-fidelity is a copy of the high-fidelity, as in this thesis case. But practically it is difficult for a human to conduct a low-fidelity test as good as a computer does with the high-fidelity. The test leader for a low-fidelity may not remember how all specific bugs behave from the high-fidelity and as a consequent forget to simulate that bug, or whatever the error is, for the user. As the test needs to be conducted very strictly and the test leader need to be well prepared, questions the feasibility in taking the low-fidelity prototype so far. As mentioned in the guidelines from last chapter, the line where low-fidelity prototyping ends and high-fidelity prototyping starts does not have one right answer. Most logical is to end the low-fidelity prototyping when it does not generate more valuable feedback. That could be after just a few evaluation iterations to several, it depends on how effective and valuable the feedback from previous evaluation results have been.

Another difficulty in context of low-fidelity prototype is counting errors. Although the usability test were both video and audio recorded, it is hard to judge whether a certain action is an error or not. And given that the test was based on having the test leader acting the computer (action performer) and performed all actions requested by the user, it was hard to judge errors from user questions like “What happens if I press there?”. Those who asked that question were encouraged to press wherever they planned to press and were also referred to not ask that kind of question due to the stated test rules. It is not unreasonable the user asks the test leader since it is the test leader who controls the most interaction and also knows what each object in the application does. Therefore the user becomes tempted, of natural causes, to engage in dialogue with the test leader. This is not likely in the high-fidelity test case, where the test leader is not a leader in the terms of controlling the test, instead more of an observer.

Also there was a problem in getting the user to remember that it was a possible to right-click, as with a mouse, in the prototype, even though the user was told that in the beginning. It may have been prevented by providing the user with a physical object for simulating a mouse. This would probably aware the user of the mouse options available, like left-, right-click or even scrolling.
8.4 Suggestions for further work

There is a never-ending debate about low- and high-fidelity prototypes equivalent efficiency in identifying usability problems in an application. This thesis discovered that there is no statistical significant difference between low- and high-fidelity prototypes, in context of usability testing, under the condition that the low-fidelity is reverse-engineered from the high-fidelity. The recommendation is to compare the efficiency in detecting errors between low-and high-fidelity prototypes, where the low-fidelity is created from scratch. In addition, that result could be compared with another set of low-and high-fidelity prototypes, with the exception that the low-fidelity prototype is reverse-engineered from the high-fidelity prototype, as in this thesis.

That will probably prove the difference of error detecting efficiency between the low- and high-fidelity prototypes even more.
References


Appendix A: Usability test

Introduction to the usability test

Det här testet är en del av testledarens examensarbete och kommer att användas i utvärderingen av applikationen Quick Builder. Testet går ut på att upptäcka problem i applikationens design och koncept. Observera att det är applikationen som ska testas och inte du.

Quick Builder finns i två olika former, med låg respektive hög naturtrogenhet. Den du har framför dig är den låga. Testet består av tio uppgifter och avslutas med en enkät.


Du behöver inte känna någon tidspress och det är när som helst under testet okej att avsluta.

Uppskattad tidsåtgång: 1 timme

Test tasks


2. Använd Quick Builder för att göra en notering i kalendern om påskkampanjen som pågår från 15 mars kl. 12.00 till och med 1 april kl. 12.00.


8. Sidan är klar nu för en första förhandsgranskning i en mobiltelefon. Hur gör du?


# Appendix B: Test results

## Transliteration of usability tests for low-fidelity prototype

<table>
<thead>
<tr>
<th>ID</th>
<th>Tid</th>
<th>Fel</th>
<th>Hintas</th>
<th>Beskrivning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID 1</td>
<td>12:47</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Letar längst ner på sidan</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Publish</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Letar efter en arkivmeny</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Settings, &quot;det här känns fel&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Graphics och vill söka på unika besökare</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Menus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Themes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Campaigns och försöker skriva in datumet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på Reports och finner svaret</td>
</tr>
<tr>
<td>ID 2</td>
<td>03:09</td>
<td></td>
<td></td>
<td>Går in på Reports och finner svaret</td>
</tr>
<tr>
<td>ID 3</td>
<td>00:57</td>
<td></td>
<td></td>
<td>Går in på Reports och finner svaret</td>
</tr>
<tr>
<td>ID 4</td>
<td>02:06</td>
<td>1</td>
<td></td>
<td>Tänker länge</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Går in på Reports och avläser först fel värde och sen rätt värde</td>
</tr>
<tr>
<td>ID 5</td>
<td>00:59</td>
<td>1</td>
<td></td>
<td>Går in på Reports och klickar på fliken år, men hittar rätt sen</td>
</tr>
<tr>
<td>Medel</td>
<td>04:00</td>
<td>1,8</td>
<td></td>
<td>Majoriteten har inga problem att hitta statistiken. Några har dock varit osäkra på om de ska kolla under fliken Year eller Week när de är inne på Reports.</td>
</tr>
<tr>
<td>ID</td>
<td>Tid</td>
<td>Fel</td>
<td>Hintas</td>
<td>Beskrivning</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Menyval</td>
<td>Annat</td>
<td></td>
</tr>
<tr>
<td>ID 1</td>
<td>00:50</td>
<td></td>
<td></td>
<td>• Går in på Campaigns och planerar kampanjen</td>
</tr>
<tr>
<td>ID 2</td>
<td>01:12</td>
<td></td>
<td></td>
<td>• Går in på Campaigns och planerar kampanjen</td>
</tr>
<tr>
<td>ID 3</td>
<td>01:49</td>
<td>1</td>
<td></td>
<td>• Går in på Campaigns och planerar kampanjen med fel namn, efter rättelse skrivs rätt namn</td>
</tr>
<tr>
<td>ID 4</td>
<td>01:09</td>
<td></td>
<td></td>
<td>• Går in på Campaigns och planerar kampanjen</td>
</tr>
<tr>
<td>ID 5</td>
<td>01:18</td>
<td></td>
<td></td>
<td>• Går in på Campaigns och planerar kampanjen</td>
</tr>
<tr>
<td>Medel</td>
<td>01:16</td>
<td>0,2</td>
<td></td>
<td>Ingen hade problem med varken att hitta Campaigns eller att planera kampanjen. Det gick förvånansvärt bra.</td>
</tr>
<tr>
<td>ID</td>
<td>Tid</td>
<td>Fel Menyval</td>
<td>Hintas</td>
<td>Beskrivning</td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>• Går in på <strong>Forms</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Menus</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Text</strong>, skriver punktlistan och lägger till den på <strong>Workspace</strong></td>
</tr>
<tr>
<td>ID 1</td>
<td>05:09</td>
<td>1</td>
<td></td>
<td>• Går in på <strong>Menus</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Text</strong>, skriver punktlistan och lägger till den på <strong>Workspace</strong></td>
</tr>
<tr>
<td>ID 2</td>
<td>03:19</td>
<td>1</td>
<td></td>
<td>• Söker med blicken efter ett verktyg för punktlistor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Menus</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Text</strong>, skriver punktlistan, men <strong>byter färg på texten</strong> och lägger sen till den på <strong>Workspace</strong></td>
</tr>
<tr>
<td>ID 3</td>
<td>05:33</td>
<td>1</td>
<td>1</td>
<td>• Vill ha någon sorts meny</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• <strong>Klickar på Templates</strong> för att importera en mall med färdig punktlista</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Text</strong>, skriver punktlistan och lägger till den på <strong>Workspace</strong></td>
</tr>
<tr>
<td>ID 4</td>
<td>04:34</td>
<td>1</td>
<td></td>
<td>• Går in på <strong>Forms</strong>, men inser snabbt att det är fel verktyg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Menus</strong>, men inser snabbt att det är fel verktyg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Forms igen</strong> och försöker dra in inputfält på <strong>Workspace</strong>, &quot;jag missförstod text-punktlistan&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Text</strong>, skriver punktlistan och lägger till den på <strong>Workspace</strong></td>
</tr>
<tr>
<td>ID 5</td>
<td>03:41</td>
<td>3</td>
<td>1</td>
<td><strong>Medel</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Majoriteten gick intuitivt in på <strong>Menus</strong>, de sa att menyikonen ser ut som en punktlista. Möjlig att samma fenomen gäller för <strong>Forms</strong>, men det är inte bekräftat.</td>
</tr>
<tr>
<td></td>
<td>04:27</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Tid</td>
<td>Menyval</td>
<td>Annat</td>
<td>Hintas</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>---------</td>
<td>-------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| 1   | 06:30 | 1       | 2     | 2      | • Vill ha en arkivmeny  
• Trycker CTRL+C och CTRL+V för att få in bild1 från skrivbordet  
• Går in på **Settings**, "då måste jag länka bilden i så fall"  
• Går in på **Graphics**, användaren vill länka bilden. Efter ett tag söker användaren efter bild1, men blir sen ombedd att browsa istället. Bilden hamnar i **Store** och det blir nu svårt att flytta den till **Workspace**, användaren säger "en add-knapp saknas". Efter lite hjälp dras bild1 och släpps på **Workspace**. Användaren säger att det skulle man testat, "när man sitter framför datorn så gör det inget om man gör fel" |
| 2   | 02:03 |         | 2     |        | • Går in på **Color**, trodde att det var ikonen för **Graphics**  
• Går in på **Graphics**, söker efter bild1, men ändrar sig och browsar fram bild1 och addar den till **Workspace** |
| 3   | 01:03 |         |       |        | • Går in på **Graphics**, browsar fram bild1 och addar den till **Workspace** |
| 4   | 02:35 |         | 1     |        | • Minimerar **Quick Builder** för att ta sig till skrivbordet och hämta bild1  
• Går in på **Graphics**, browsar fram bild1 och addar den till **Workspace** |
| 5   | 01:33 |         |       |        | • Går in på **Graphics**, browsar fram bild1, drar och släpper den på **Workspace** |
| **Medel** | 02:45 | 1,2     |       |        | En kommentar till den första användarens länkning av bild: 
Användaren, som är journalist, berättar att när man behöver en bild i sin artikel då brukar man länka den från en server. I dennes arbete importeras inte bilder på samma sätt som i Quick Builder. I övrigt är det inte några problem att lägga in en bild förutom att några ville söka efter bilden, vilket givetvis ska fungera då funktionaliteten erbjuds, men att den i dagsläget saknar underliggande kod |
<table>
<thead>
<tr>
<th></th>
<th>Tid</th>
<th>Fel</th>
<th>Hintas</th>
<th>Beskrivning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Menyval</td>
<td>Annat</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>1</td>
<td>för att faktiskt utföra själva sökning.</td>
</tr>
<tr>
<td>ID 1</td>
<td>02:1</td>
<td>4</td>
<td>1</td>
<td>• Går in på <strong>Color</strong>, använder den för att färga texten, men <strong>använder fel tillvägagångssätt</strong>. Blir ombedd att dra och släppa färgerna på texterna.</td>
</tr>
<tr>
<td>ID 2</td>
<td>02:1</td>
<td>7</td>
<td></td>
<td>• Går in på <strong>Text</strong>, markerar texten och använder textfärgen för att byta färg.</td>
</tr>
<tr>
<td>ID 3</td>
<td>01:2</td>
<td>4</td>
<td>1</td>
<td>• Går in på <strong>Text</strong>, markerar texten, använder textfärgen för att byta färg och trycker på add-knappen.</td>
</tr>
<tr>
<td>ID 4</td>
<td>02:3</td>
<td>5</td>
<td>1</td>
<td>• Går in på <strong>Text</strong>, text-color, förstår först inte att texten måste markeras för att redigera den och trycker därför på den med avsikt att markera den.</td>
</tr>
<tr>
<td>ID 5</td>
<td>01:5</td>
<td>6</td>
<td>1</td>
<td>• Går in i <strong>Source</strong>, men blir ombedd att leta någon annanstans.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Går in på <strong>Text</strong>, vill dra och släppa till textredigeringsrutan, men gör rätt sen.</td>
</tr>
<tr>
<td>Medel</td>
<td>02:0</td>
<td>5</td>
<td>1</td>
<td>Här gjordes lite olika fel. Felen kan sammanfattas med att användarna inte vet hur just Quick Builder beter sig och att text automatiskt uppdateras i <strong>Workspace</strong> när man redigerar den i textrutan i <strong>Text</strong>.</td>
</tr>
</tbody>
</table>

**Medel:**

<table>
<thead>
<tr>
<th>Tid</th>
<th>Fel</th>
<th>Hintas</th>
<th>Beskrivning</th>
</tr>
</thead>
<tbody>
<tr>
<td>02:0 5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Tid</td>
<td>Fel</td>
<td>Hintas</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>ID 1</td>
<td>02:1 6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ID 2</td>
<td>02:1 3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ID 3</td>
<td>00:0 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 4</td>
<td>00:2 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 5</td>
<td>00:1 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medel</td>
<td>01:0 2</td>
<td>0,8</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Tid</td>
<td>Fel</td>
<td>Hintas</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>ID 1</td>
<td>01:57</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 2</td>
<td>03:25</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 3</td>
<td>05:40</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 4</td>
<td>01:43</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID 5</th>
<th>03:4 4</th>
<th>6</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Går in på Graphics</td>
<td>Drar och släpper bild1 utanför Workspace i hopp om att radera den vid två tillfällen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Raderar bild1 ur Store i hopp om att den ska försvinna från Workspace</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Markerar bilden och trycker på delete-knappen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Söker först efter bild 2 och sen browsar, drar och släpper den på Workspace</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Markerar bilden och trycker på cancel i Graphics</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Efter tips högerklickar användaren på bild1 och väljer delete</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Medel</th>
<th>03:1 8</th>
<th>3,6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Tid</td>
<td>Fel</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>-----</td>
</tr>
<tr>
<td>ID 1</td>
<td>01:5</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 2</td>
<td>04:2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 3</td>
<td>04:4</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 4</td>
<td>05:2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID 5</td>
<td>00:3</td>
<td></td>
</tr>
</tbody>
</table>

Medel 03:2       4

| Tre användare tycker att Publish-ikon ser ut som och att Publish låter som om det vore en Preview-knapp. Samma användare gick in på Settings och ställde in mobilnumret. Några nämnde att Preview-knappen var svår att hitta, dels för att den inte låg där den förväntades ligga och dels |
förr att den tillsammans med andra ikoner blev så pass hoptryckta och grötiga att de inte syntes.

<table>
<thead>
<tr>
<th>9</th>
<th>Tid</th>
<th>Fel</th>
<th>Hintas</th>
<th>Beskrivning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Menyval</td>
<td>Annat</td>
<td></td>
</tr>
<tr>
<td>ID 1</td>
<td>05:46</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
|   |      |      |      | • Blickar över **Workspace**-ikonerna  
|   |      |      |      | • Trycker på **Save i Workspace**  
|   |      |      |      | • Trycker på **Preview**  
|   |      |      |      | • Trycker på collapse för **Workspace**  
|   |      |      |      | • Högerklickar på **Workspace** och väljer properties  
|   |      |      |      | • Högerklickar på sidhuvudet för **Workspace**  
|   |      |      |      | • Högerklickar tillslut på bild2 och väljer set visibility  
| ID 2 | 00:46 | 1 |      |             |
|   |      |      |      | • Högerklickar på bild2 och väljer properties  
|   |      |      |      | • Högerklickar på bild2 och väljer set visibility  
| ID 3 | 00:34 |      |      |             |
|   |      |      |      | • Högerklickar på bild2 och väljer set visibility  
| ID 4 | 02:15 | 1 |      |             |
|   |      |      |      | • Markerar bild2 och letar efter inställningar i **Graphics**  
|   |      |      |      | • Högerklickar på bild2 och väljer set visibility  
| ID 5 | 02:01 | 3 | 2 |             |
|   |      |      |      | • Går in på **Graphics** och trycker på alt  
|   |      |      |      | • Går in på **Settings**  
|   |      |      |      | • Går in på **Themes**  
|   |      |      |      | • Högerklickar på bild2 och väljer properties  
|   |      |      |      | • Högerklickar på bild2 och väljer set visibility  
<p>| Medel | 02:16 | 2,6 |      | Tre av användarna tänker att de ska hitta de kundanpassade inställningarna i properties för bilden. |</p>
<table>
<thead>
<tr>
<th></th>
<th>Tid</th>
<th>Fel</th>
<th>Hintas</th>
<th>Beskrivning</th>
</tr>
</thead>
</table>
| ID 1 | 13:58 | 7   | 2      | - **Går in på Text**, trodde att texten popStarRecords.gif skulle skrivas **Forms**.  
- **Går in på Forms**, trodde det var **Graphics**.  
- **Går in på Graphics**, browsar efter popStarRecords.gif, drar och släpper den på **Workspace**.  
- **Går in på Menus**, skriver i länknamn och addar länkarna till **Workspace**.  
- **Går in på Text**, skriver Welcome to popstar records, addar den till **Workspace**.  
- **Går in på Services**, drar och släpper tjänsten Most downloaded på **Workspace**.  
- **Går in på Graphics**, browsar efter AliciaKeysBanner.gif, drar och släpper den på **Workspace**.  
- **Går in på Forms** igen.  
- **Går in på Settings**, ”inställningar, jag tänker om jag ska utanför den sidan så känns det som att man måste koppla den med en länk”.  
- **Går in på Menus** igen.  
- **Efter tips** går användaren in på **Text**, skriver Visit Alicia Keys och addar den olänkad till **Workspace**.  
- **Går in på Menus** igen.  
- **Efter tips** hittar användaren länkningsverktyget och länkar texten till www.aliciakeys.com. |
| ID 2 | 06:00 | 1   | 2      | - **Går in på Graphics**, hittar popStarRecords.gif i **Store**, drar och släpper den på **Workspace**.  
- **Går in på Menus**, skriver fel länknamn, tillrättavisas och addar varje länk till **Workspace**.  
- **Går in på Text**, skriver Welcome to popstar records, addar den till **Workspace**.  
- **Går in på Services**, användaren konstaterar ”det verkar kosta pengar”, drar och släpper tjänsten Most downloaded på **Workspace**.  
- **Går in på Graphics**, hittar AliciaKeysBanner.gif i **Store** och drar och släpper den på **Workspace**. |
<table>
<thead>
<tr>
<th>ID</th>
<th>04:4</th>
<th>1</th>
</tr>
</thead>
</table>

- Går in på **Text**, skriver Visit Alicia Keys och addar den olänkad till **Workspace**
- Högerklickar på texten och väljer **properties** i hopp om att hitta verktyg för länkning
- Går in på **Text**, efter tips hittar användaren länkningsverktyget och länkar texten till www.aliciakeys.com

<table>
<thead>
<tr>
<th>ID</th>
<th>08:1</th>
<th>7</th>
</tr>
</thead>
</table>

- Går in på **Graphics**, hittar popStarRecords.gif i **Store**, drar och släpper den på **Workspace**
- Går in på **Menus**, skriver i länknamn och addar varje länk till **Workspace**
- Går in på **Text**, skriver Welcome to popstar records, addar den till **Workspace**
- Går in på **Services**, drar och släpper tjänsten Most downloaded på **Workspace**
- Går in på **Graphics**, hittar AliciaKeysBanner.gif i **Store** och drar och släpper den på **Workspace**
- Går in på **Text**, skriver Visit Alicia Keys, markerar texten, hittar länkningsverktyget och länkar texten till www.aliciakeys.com

<table>
<thead>
<tr>
<th>ID</th>
<th>09:3</th>
<th>8</th>
<th>4</th>
</tr>
</thead>
</table>

- Går in på **Graphics**, hittar popStarRecords.gif och AliciaKeysBanner.gif i **Store**, drar och släpper dessa på **Workspace**
- Går in på **Menus**, addar varje länk till **Workspace**
- Går in på **Text**, skriver Welcome to popstar records, addar den till **Workspace**
- Går in på **Menus** och planerar att göra en punktlista som ser ut som den i skissen
- Går in på **Text** och planerar att göra en punktlista som ser ut som den i skissen
- Går in på **Services**, addar tjänsten Most downloaded
- Byter plats på elementen i **Workspace**
- Går in på **Menus** och planerar att göra en länk
- Går in på **Text**, skriver Visit Alicia Keys och addar den olänkad till **Workspace**
- Högerklickar på texten och väljer **properties**
- Går in på **Source**, i hopp om att kunna koda länken
- Går in på **Services**
- Går in på **Publish**
- Går in på **Settings**
- Går in på **Text**, hittar länkningsverktyget och länkar texten till www.aliciakeys.com

<table>
<thead>
<tr>
<th>Medel</th>
<th>08:3</th>
<th>4,2</th>
</tr>
</thead>
</table>

Några användare hade problem med att hitta länkerverktyget som ligger i **Text**. Det förekom också en viss osäkerhet över skillnaden på en textlänk och en meny, då båda kan användas som länkar.

---

**Transliteration of usability tests for high-fidelity prototype**

Appendix C: Questionnaire

Comparison chart between questionnaire results from low-and high-fidelity prototype

The bars show the median rating from both the low- and high-fidelity group. A rating reaches from 1 to 4; where 1 means strongly disagree, 2 disagree, 3 agree and 4 strongly agree.

1. It is difficult to get around in the application.

2. I can quickly find what I’m looking for in this application.

3. This application seems logically structured.
Appendix C

This application would need more explanation, initially.

I think the application’s look is very appealing.

I feel like I’m in control when using the application.

Learning how to locate things in this application is problematic.

This application helps me to perform the tasks I want.
Appendix C

I feel frustration when I use this application.

<table>
<thead>
<tr>
<th></th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

I miss one/several features of this application (if so, what / which?).

<table>
<thead>
<tr>
<th></th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>