Evaluating GUI design without users

Interaction Design 180 ECTS credits

Therése Andersson, therese.anderssons@gmail.com
Abstract

This work has been performed at Ericsson in Linköping, and it has been focused on one of their applications, Event Based Statistics (EBS). The goal of this work was to design a new GUI proposal for EBS which later could be implemented by the programmers at Ericsson.

Most of the developers have no direct contact with the end users, since the applications go through many test stages before they reach the customers. This became a problem as the core of Interaction Design is about designing according to users’ needs and requests.

The purpose of this report was to investigate how the methods of design evaluation without users worked and to analyze each method’s benefits and disadvantages. The result of the practical work became a prototype which visualizes how the detected usability problems have been solved. The prototype was developed in Java by the help of the GUI design tool in Netbeans. The parts which could not be implemented in the prototype because of technical or time constraints were explained in text in the report. The scientific investigation four methods were implemented to evaluate the usability of the interface. The methods were carried out and evaluated and were finally compared with each other.

Keywords: Usability evaluation, graphical interface, method comparison.
Preface

This is an examination thesis for the University of Kalmar, School of Communication and Design. The work has been running over a period of ten weeks. As a reader of the report no technical background knowledge is needed, although the interest in the content increases if people are interested in usability and interface design.

My personal goal of the thesis was to gain a better understanding of how user friendly interfaces are created and thus be able to clarify what is required to achieve good user-centered design in the development of a system. Another objective was to find which evaluation methods that work well in practice when designing graphical interfaces when no end user feedback is available.

I want to give a cordial thanks to the EBA team at Ericsson for friendly treatment and the assistance to carry the work forward. The commitment of this team and their interest in the work has been very helpful. Thanks also to my supervisor at University of Kalmar, Hanna Broberg, for advice and constructive criticism during this work, and to my supervisor at Ericsson, Daniel Franzén.

Therése Andersson

Linköping, May 2009
# Contents

Abstract ................................................................................................................ I

Preface .................................................................................................................. II

Contents ............................................................................................................ III

1. **Introduction** ............................................................................................... 1

2. **Background** ............................................................................................... 2

   2.1 Ericssons’s operational and maintenance systems ...................................... 2

   2.1.1 Event Based Statistics ............................................................................ 3

   2.2 Usability studies ........................................................................................ 4

   2.3 Problem definition and question formulation .............................................. 6

      2.3.1 Purpose ............................................................................................... 6

      2.3.2 Objective ............................................................................................ 6

      2.3.3 Delimitations ....................................................................................... 7

3. **Theoretical starting points** ........................................................................ 8

   3.1 Methods of design evaluation without users .............................................. 8

      3.1.1 Cognitive Walkthrough ..................................................................... 8

      3.1.2 Action Analysis ................................................................................. 9

      3.1.3 Heuristic Evaluation ......................................................................... 10

         *Fel bokmärket är inte definierat.*

      3.1.4 Focus Groups .................................................................................... 11

   3.3 A comparison of techniques for evaluating design .................................... 12

   3.4 Design Principles for creating appropriate interfaces .............................. 13

4. **Methods** .................................................................................................... 15

   4.1 Methodological considerations ................................................................. 15

   4.1.1 Approach ............................................................................................ 15

   4.1.2 Research ............................................................................................. 15

   4.2 Approach and method .............................................................................. 16
4.2.1 Action research
4.2.2 Observation of method application
4.2.3 Method analysis
4.2.4 Implementation of Heuristic Evaluation
4.2.5 Implementation of the Focus Groups
4.2.6 Implementation of Cognitive Walkthrough
4.2.7 Implementation of Action Analysis
4.2.8 Prototype construction

5. Results
5.1 Results of Heuristic Evaluation
5.2 Results of Focus Groups
5.3 Results of Cognitive Walkthrough
5.4 Results of Action Analysis
5.5 The new design proposal
5.5.1 Application of different frames
5.5.2 Poor overview
5.5.3 Poor search function
5.5.4 No copy function
5.5.5 A slow way to create new counters and to change them
5.5.6 No information about counters
5.5.7 Proposals that were not included in the new design

6. Discussion
6.1 Performance Problems
6.2 Methods
6.2.1 Critical examination of Focus Groups
6.2.2 Critical examination of Heuristic Evaluation
6.2.3 Critical examination of Cognitive Walkthrough
6.2.4 Critical examination of Action Analysis
1. Introduction

The graphical user interface is celebrating its 25th anniversary this year if one defines the launch of the first Mac in 1984 as its birth. When the files became icons that resembled sheets of paper, and directories became folder icons, it became easier for new users to understand how they could use the computer (Nielsen, 2009).

As technology evolved, user-friendliness has become increasingly important (Cooper, 2007), so even at Ericsson, which traditionally has been a technology-oriented company. One of the applications in Ericsson’s operational and maintenance systems which suffer from poor usability is Event Based Statistics, EBS. Some users have chosen to use Microsoft Excel instead of the interface in EBS because it has been cumbersome and time consuming to use. Most developers at Ericsson have no contact with the end users, which led to an interesting problem: Which methods for evaluating usability are available if you do not have direct contact with the users?

The practical task was to design a new interface proposal for EBS, in the form of an executable prototype. The scientific work has been to use methods to find usability shortcomings in the interface and to evaluate the methods by comparing them in terms of time consumption, resource needs and results.
2. Background

The background is intended to provide understanding of the emergence of the problem and to describe how the application in focus works. In addition, it also includes an explanation of the concept of usability studies and the importance of this to achieve successful interface design.

2.1 Ericsson’s operational and maintenance systems

The thesis work has been performed at Ericsson in Linköping, at the Development for the operation and maintenance system. The system is called Operations Support Services (OSS) and it runs on computers with the Solaris operating system. The unit is divided into several sections, the two largest are located in Linköping and in Ireland. The system is used by mobile network operators to control and monitor their mobile networks.

One of the product families in OSS is Event Based Application (EBA) which collects information from the network in the form of small data reports called events. Examples of events are when a mobile phone call is started, ended or aborted. The application Event Based Statistics (EBS) is available in three variants; one for the older GSM mobile radio technology, one for WCDMA (which in everyday language is known as 3G) and one for the core telephone network (used when dialing with a regular wall phone). All three variants have the same functionality and interface. The only thing that differs is the set of available counters that the user can choose from. A counter counts the number of events during a certain period of time in a certain part of the mobile network. An example is the number of dropped (aborted) calls during the last 15 minutes. The counter values are either written to an XML file or are stored in a database.

As the graphical interface of EBS was considered difficult to understand and use, the task was to review the usability of this defined part of the EBA product family and make a new design proposal in which the identified usability deficiencies were resolved. The application should of course keep its existing functionality, but apart from this there were no additional requirements on the design or how the interface should look like or be structured.

The results of this thesis should be a design proposal on what the new GUI can look like, in the shape of an executable prototype that clearly shows the design ideas.
2.1.1. Event Based Statistics

The existing interface is shown in Figure 1. The window is divided into three frames. The frame to the left, with a “Monitors” header, contains all available counters that the user can choose from. They are divided into logical units; for example, all counters that have to do with speech quality are in the folder “CS Quality”. The upper right frame, “Definition”, contains all the counters that the user has selected. To add a new counter, the user drags a counter icon from the Monitors frame and drops it onto the Definition frame. It is also possible to add a counter by double-clicking it. Counters can be put in different folders and each folder is called a counter group. If the user drags and drops, one can choose the folder in which the counter will be placed, but if the user double-clicks the counter will always end up in the folder at the bottom.

The lower right frame contains the settings for the selected counter. Here it is possible to set up filters for it. Filters are limits on what the counter should count. For instance, assume that a counter counts the number of dropped calls and that it has a filter with four values: Sony Ericsson, Samsung, Nokia and Other. If no filter is selected the counter will count all dropped calls, but if the filter is set to Nokia only dropped calls from Nokia phones will be counted. In addition to these standard filters there is a feature called Distribution that creates a counter for each filter value combination. If Distribution is selected for the counter in the previous example four counter instances will automatically be created; one that counts dropped calls from Sony Ericsson phones, one from Samsung phones, one from Nokia phones and the last from all remaining manufacturers.

There are menu items to export the counter definition to an XML file, which later can be imported.

Counter definitions can be very large and a typical definition consists of thousands of counters. These will then report values for different parts of the network, so the number of expected values per quarter of an hour in the XML files or the database might be several millions. Creating counters and applying filters in the GUI is a cumbersome and time consuming process. There is also no information in the interface about how many counters that has been created. These parts needed to be improved.

That which was known about the application’s users was that they were mobile operators and were located all over the world, but not in Sweden. The largest
customer was AT&T in the United States. It also had emerged that they used Excel to create definitions which they later imported into the EBS GUI. More detailed information about the users was not available.

![The Event Based Statistics window.](image)

**Figure 1.** The Event Based Statistics window.

### 2.2 Usability studies

Usability (Ottersten and Berndtsson, 2002) is a quality characteristic of interactive products. The product has high usability if it meets the originator’s and the user’s requests. A product’s usability is revealed in the interaction between the product and its users over a period of time. Nielsen (1999), one of the world’s leading experts in user-friendliness of interfaces, means that a general principle for all interface design is that simplicity beats complexity.

A good way to reach simplicity is to remove all design elements that do not contribute to the design or give any value to the users. Nielsen believes that an interface designer must facilitate and support user-controlled navigation. It is about not forcing the user into a given condition but to allow freedom of movement. In traditional interface design it is the designer who determines what options that should be available, for instance by disabling commands that are not appropriate in the context.
Cooper (2007), thinks that usability means that the interface should be so easy to use so that the users interact with the system without having to think about it. This means that the interaction should feel natural and that all operations should be so intuitive that users understand how they are supposed to be used without any additional instructions.

There are two major groups of evaluation methods of usability: audit-based evaluations (conducted by the interaction designer on her own) and user-based evaluations (where users are involved through their behavior in the system which is studied and their views are noted in interviews) (Nielsen, 2009). These studies can be carried out at different times during a development process, on a prototype or on a finished product. Peer reviews are conducted by an evaluator with expertise and the purpose is to find usability problems with little effort. After a peer review the shortcomings, and possible solutions, are outlined. In particular, attention is put on lack of consistency and unclear or confusing interactions.

User tests as a method is about measuring the user friendliness and to ensure that the system works as intended. The tests consist of test tasks and are performed on real users during relevant conditions. One observes how the user navigates and uses the system to perform the test tasks. The result of the method provides a list of measures, in priority order, that should be corrected in order to give the system the desired effect (Nielsen, 2009).

What should be evaluated differs between different systems and types of web sites. It may include information architecture, language and communication if it is about an information site or efficiency and minimization of errors in a work-supporting application. Today mobile solutions are also popular and then the interaction opportunities and the customer experience are in focus. To test an interface on users in an evaluation process has several advantages for the final design.

Hobart (1995) has formulated common design mistakes where the problem is that the user is not involved. He suggests that developers often make their design decision from what they know, not what the users know. There are also those who believe that audit-based evaluations can identify the same number of short-comings, or even more, compared to user-based.
2.3. Problem definition and question formulation

The problem of the study was that there was no opportunity to obtain information or feedback directly from end users. There was information available about the problems that users experienced in the existing application, but there was no opportunity to evaluate the new design on end users. As a complement to the problem formulation, research questions were presented. They were formulated on the basis of the problem and were the focus of the evaluation. Also, there were questions that were to be answered after the evaluation had been completed.

What methods are appropriate to use when you do not have access to users in a design process?

What methods to evaluate the usability of the interface proved to be most successful and what were the benefits and disadvantages of each method?

2.3.1 Purpose

The purpose was to develop a new design proposal of the interface for EBS, and to use and evaluate the methods that were available when one does not have direct contact with users. To address the research problem, four methods were going to be used which are said to be suitable for design evaluation without access to users. The idea was to examine how well the methods worked. The methods were compared with each other on the basis of three aspects; what was required to perform the method, how much time had been spent on planning, implementing and reviewing the method, and the quality of the finished result. This means the strengths and weaknesses found after the implementation of the method and which results that turned out to be the most valuable for the work on the new design proposal.

In conclusion, the method is compared against the following aspects:

- Resources (in terms of people)
- Time
- Results

2.3.2 Objectives

The objectives of the new interface are to provide an improvement by presenting a number of identified problems in the current interface, evaluate them and then create
a prototype that presents a solution to each identified problem. The decisions must be justified and any returns to fix the problems should be presented. Starting from the functions that are considered to violate usability, or that are completely missing, the goal is that new design should make it easier for existing users to use and understand the application. With regard to the problem that users today are using Excel instead of the available interface for a more efficient way of working, the goal with the new design is that it should no longer be necessary. The purpose of this thesis is also to clearly explain how the design methods that were used worked, what they accomplished and to demonstrate their advantages and disadvantages.

2.3.3 Demarcations

The work was restricted to a part of the application EBA, namely Event Based Statistics (EBS). This feature was considered to be the main problem area when it came to usability and was also a feature which dimension and complexity was considered reasonable for the time schedule of the work. Technical limitations were not taken into account. Another demarcation line was that there was no requirement to address all problems in the design proposal. Functions which were impossible to implement in the prototype due to technical limitations in the used tool (Netbeans), or could not be implemented due to time constraints, should be presented in descriptive text. There were also a number of found methods to evaluate usability of interfaces, and four of them were chosen.

To summarize the above, the following delimitations are valid:

- Only the statistics window of the EBS application
- Address problems that were possible to visualize in a prototype
- Number of evaluation methods used
3. Theoretical starting points

The purpose of the literature study was to create an understanding of the evaluation methods that were to be used. It was also relevant to look at similar assessments, and also what it takes to succeed in creating a user-friendly interface.

3.1 Methods of design evaluation without users

Lewis and Riemann (1994) mention three possibilities to evaluate an interface with the absence of users. The three methods discussed are cognitive walk through, Action Analysis and Heuristic Evaluation. Lewis and Riemann believe that a good design evaluation without the involvement of users may discover the same number of problems that an evaluation with only a few users may discover. An interface can be used by thousands of users and tests with only a few of them will not disclose all the problems. Also, some of the problems will only arise after some time when the users have gained more experience. An evaluation without users does not reveal all of the problems; the chances of success are greater if both types of evaluations are applied.

3.1.1 Cognitive Walkthrough

Cognitive Walkthrough (Lewis and Riemann, 1994) is a method to imagine the users’ thoughts and actions when using the interface for the first time. This is based on a prototype or a detailed design description of the interface. A prerequisite is also to know who the users are. An evaluation is carried out, broadly speaking, by formulating a couple of tasks that have to be solved by using the prototype. The evaluators think aloud how they think a user would reason when they perform the tasks. If the design of the interface is not properly supporting the tasks, it is possible to identify where the problems are located and to correct them by iterative design. The most successful tests are made by designers who have worked close to real users as they can create a mental picture of the users in their current environments. If a design change has to be made in solitude, a mental cognitive walk through can be performed to check the design. After that the major parts of the interface have been settled, it is useful to make a walk through with a group, for instance with other designers and users. One thing to remember is that the method works as a tool to develop and improve the interface, not to validate it. The obtained results are used to correct the interface and a lot of the changes may sound obvious, such as making GUI components more visible, use terms
that the users are familiar with and to provide users with better feedback. Before a walk through is carried out information about and an understanding of the following four items are needed:

- **A description or a prototype of the interface**
- **Task description**
- **A complete, written list of the actions needed to complete the task with the interface**
- **An idea of who the users will be and what kind of experience they will bring to the job**

While the stories are shaped, it is recommended to bear the following things in mind:

1. **Will the users be trying to produce whatever effect the action has?** Users do not always think what designers expect them to think.

2. **Will users see the control (button, menu etc) for the action?** It is more important to discover the control than to identify the purpose of it.

3. **Once users find the control, will they recognize that it produces the effect they want?** Even if users want to perform the correct action and the control is visible, will they figure out that this is the right control?

4. **After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?** Even simple actions need some sort of feedback to demonstrate that the system responded to the action. Different types of feedback apply to different types of systems. For example, a green light indicates that a printer is turned on, and an icon can be highlighted when navigating in a graphical interface.

### 3.1.2 Action Analysis

This method (Lewis and Riemann, 1994) provides accurate predictions for the time it takes an experienced user to complete a task. Since the objectives of the application are identified, the information about what the application will deliver and contribute to the users is known. To predict the time it is estimated how long it takes, both physically and mentally, to carry out any small part of a task. Many of the steps take only a fraction of a second, such as pressing a key on the keyboard or moving the eyes to a new location on the screen.
Defining the task in an Action Analysis is an iterative process. The basic tasks are divided into subtasks, and then each of those subtasks is broken into even smaller subtasks, until the description reaches fraction-of-a-second level operations. This method is used to examine whether some functions need to be reviewed and to reduce the number of steps needed to perform a task.

3.1.3 Heuristic Evaluation

Nielsen (Lewis and Riemann, 1994) has developed a method to inspect the user interface. The method is known as Heuristic Evaluation and is a form of peer review, which was developed about 15 years ago. The method aims at finding usability problems in a user interface to be able to address them in a design process. The evaluation consists of ten general guidelines, also called heuristics, describing what an interface should meet to be regarded as user-friendly. The system is then examined by an expert who registers all “crimes” against the guidelines. The problems are ranked in order of seriousness and then compiled in a report. The expert should evaluate the system in between one and two hours. Would a system be so large or complex that an expert needs more time, the evaluation should be divided into several different occasions, rather than that the evaluation drags out for several hours.

Below are Jakob Nielsen’s ten guidelines which an interface should meet.

1. Visibility of system status. The system should always keep the user informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world. The system should speak the user’s language, with words, phrases and concepts which are known to the user. Information should be presented in a natural and logical order.

3. User control and freedom. Users often make mistakes, and therefore require clearly visible “emergency exits” to undo the last executed action.

4. Consistency and standards. Users should not need to consider whether different words, situations, and features mean the same thing. The systems should follow a uniform standard.

5. Error prevention. Even better than emergency exits and error messages is a carefully constructed design that prevents that the error occurs.
6. Recognition rather than recall. Objects, features and options should be visible. The user should not need to remember information from one part of the system to another. Relevant functions should be visible or replaceable all the time.

7. Flexibility and efficiency of use. The system will accommodate both experienced users and beginners, by letting them choose how quickly they will use the system.

8. Aesthetic and minimalist design. The interface should not contain irrelevant information. For any new information added, the competition becomes greater for the relevant items, and their visibility is reduced.

9. Help users recognize, diagnose and recover from errors. Error messages should be expressed in simple language (no codes). They should explain the problem in an exact way and suggest solutions.

10. Help and documentation. Even if a system ideally should be used without documentation, it may be necessary to provide help and documentation. All help and documentation should be easy to search in, list concrete steps and not be too wide in its extent.

3.1.4 Focus Groups

Wibeck (2000) discusses what is meant by Focus Groups and how and when this type of interview method is applicable (Wibeck 2000). Focus Groups are a form of group interviews, for which a small number of people, six to eight participants are recommended, where a given topic is discussed. The word focus indicates that the discussion should involve a pre-defined topic. The interview is led by a conversion conductor, a moderator, who will lead the discussion and introduce new aspects of the subject when it is necessary. Every interview occasion usually takes between one to two hours. The moderator is not an interviewer, but the purpose of the Focus Group is that the participants can discuss freely with each other. Focus Groups as a method is used to examine how different views, attitudes and ideas are being expressed in a group. Besides serving the purpose of data collection for research purposes, the interaction itself is studied; how the participants reason and how they come to a common conclusion, or how they come to an agreement that they have different views. They are also focused since the participants have things in common; in this case the participants all worked at the same workplace and they had great knowledge about the topic. This type of affinity among the participants meant that they talked more
freely without having to feel left out or inferior to other members. The method can be applied at any time during the study, but is especially valuable if it is just at the beginning.

3.3 A comparison of techniques for evaluating design

Doubleday, Ryan, Springett and Sutcliffe (1997) have made a comparison of two different techniques that are available when design should be evaluated from the aspect of user-friendliness. The comparison was made between Heuristic Evaluation and user tests. The purpose with the evaluation was to examine the problems that people encounter when carrying out information-based tasks and to evaluate the evaluation methods themselves by looking at three aspects: the problem focus, the quality of the results and the cost of each method. Moreover, the strengths and weaknesses of the methods were graded. The aim of the study was also to help designers to determine the most cost effective way to evaluate the interface. The evaluation result was that five experts identified 86 user problems while 20 end-users identified 38 problems. The authors point out that the definition of what an “error” is is imprecise and can be expressed in different ways between expert evaluators and other evaluators. The dividing line between different faults can be fluid and overlapping; the defects identified by a person can include various symptoms of different separate errors. A complex error can reduce the total error number, but can hide other error scenarios. It is also clear that the more errors an interface has, the more it is to find, but the most the important and most difficult measurement is how few errors the evaluator have missed. In total 33.5 hours were spent on the Heuristic Evaluation and 125 hours on the user testing. Ann Doubleday, Michele Ryan, Mark Springett and Alistair Sutcliffe think it is important to resolve as many as possible of the user problems, but from the user’s point of view, the quality of the improvements and the seriousness of the problems are the main things. An interface that produces the wrong answer or prevents the user’s activity is much worse than one that only requires more precise movement with the mouse, since the latter easily can be learnt. The problems are divided into different severity levels:

- **High**: if the user cannot continue without help, or performs the task in a wrong way.
• Medium: if the user continues after a moment of doubt, or tries to find alternative ways to accomplish the task.

• Low: if the user continues with some doubt and thoughtfulness.

In performed experiments 39% (15/38) of the usability problems found in user tests had not been found by heuristic evaluators. Approximately 40% (34 of 86) of the problems discovered by Heuristic Evaluation were not discovered by user tests. This shows that both types of evaluations are necessary. It should be noted that there is a difference between how the errors are reported between the two methods, since the experts look at the causes of errors and predict the effects, while the end-users encounter the errors.

3.4 Design Principles for creating appropriate interfaces

Nielsen's design guidelines (Lewis and Riemann, 1994) have the objective to evaluate the usability but there are guidelines that will serve more as a guide for achieving good and user-friendly interface. Design principles are often obtained by experience and can be applied to most interactive systems. It is then important to know what is required for this. There are many proposed design principles on how to succeed in interface design. Hobart (1995) and Shneiderman (1998) present a number of design principles which serve as guidelines that should be taken into account in the development of a new interface. Design guidelines are a mix of theoretical knowledge, experience and common sense. The purpose of the guidelines is to provide suggestions on what should be included and what should be avoided in the design, simply to help the designer to take the right design decisions. Guidelines can be especially useful to follow when there is no usability expertise in the design team (Shneiderman, 1998). The main characteristic of a successful interface is intuitivity. One way to achieve this is to, whenever possible, use metaphors taken from the real world. A metaphor can be a MasterCard or Visa button to pay bills in a web shop. This resembles of how the user would have paid in a real shop, and helps the user to understand the interface. Shneiderman (1998) has set up a rule that it is good to enable shortcuts for frequent actions. As the frequency of use increases, so does also the user’s desire to reduce the number of steps required to perform the action. Knowledgeable users often appreciate abbreviations, shortcuts, hidden commands, and more. Short response time is also
desirable for frequent users. Other properties that an interface should possess, according to Hobart, is speed, or rather responsiveness. The feedback is also important. Hobart also presents guidelines on how many user controls one “may” use. Shneiderman thinks that dialogues should be designed to produce a conclusion. This means that informative feedback on the closure of a group of actions gives the user a feeling of satisfaction and allows them to prepare for the next group of actions.

Another guideline of Shneiderman says that the system should be designed so that users cannot make serious errors. Prefer, for example, menus rather than forms and do not allow letters in boxes that are intended for numbers. If the user makes an error, the system should detect this and provide a simple, constructive and specific explanation of how to restore the system. Users should for example not have to rewrite an entire command, but only the part that is wrong. Erroneous actions should leave the system in an unchanged state, or the system should give instructions on how to repair the damage and restore the system. Actions should be possible to undo as to a large extent as possible. This allows the user to not worry about making mistakes, since he or she knows that the errors can be reverted and encourages in this way the user to explore new alternatives (Shneiderman, 1998).
4. Methods

This chapter describes the approach and the methodology that was determined to be used in the work. The method chapter also contains descriptions of the methodological considerations that have been made. The purpose is thus to give the reader an understanding of the way to the results.

4.1 Methodological considerations

This section presents the considerations made in the investigation, such as the choice of approach and research method.

4.1.1 Approach

A scientific study can have an inductive or a deductive approach (Johasson, 2003). An inductive approach means that research is based on empirical data, and from this information attempts are made to find theories that explain the results. It can be described as an exploratory path in the research process. Deductive approach to research is based upon existing theories, from which new subordinate hypotheses can be derived. It means a way to prove or reject existing theories. A third approach, called the abductive approach, can be seen as a combination of the inductive and deductive approaches. This means that the research is based on both empirical data and existing theory.

This study is based on a deductive approach, since there are established theories that are relevant to the investigation and the problem area. A deductive approach is to examine assumptions (hypotheses) against experience (theory). In this case, the evaluation methods will be tested to see if the theory is consistent with reality.

4.1.2 Research

Research method can be either qualitative or quantitative (Nyberg, 2000) and the fundamental difference between these is that quantitative methods transform information into figures and numbers, while qualitative methods are the researcher’s perception or the interpretation of information is at the forefront. Qualitative research method was used in this work. When using a qualitative approach the researcher’s interest is in the context, structure, and understanding rather than in explanations.
Qualitative methodology is intended to provide an overview of the whole situation and is characterized by the closeness that exists between the study and data collection. Its objective is to give the reader a deep understanding of the investigation. The thing in focus is to make a deeper study and not to have a width of the test materials, which quantitative studies rather could do. They are interested in several variables, descriptions and explanations. As the qualitative approach provides the opportunity to see the context at a deeper level, it is the chosen method. To some extent quantitative methods also have been used to answer questions like: How often, and to what extent? One example is when the priority in the Heuristic Evaluation was defined.

4.2 Approach and method

This section is an explanation of how action research has been applied and also the approach of each evaluation method.

4.2.1 Action research

To give an insight into the basis of the examination work, it will be explained what action research is and how it has been applied. The purpose of action research has always been to solve problems and to increase scientific knowledge (Broman, personal communication, 2009-02-16). It is about developing and improving a practical activity but also to gain knowledge of how this change works and what happens during this work. Action research consists of the concepts of action, research and learning. Actions are about improving something by an active user action. Research is about to systematically follow the documentation and learning to provide new insights by following and execute the work. Action research is a method in design research and represents the meeting between two practices: Research practice and operational practice. The research practice means that a scientific problem is studied by a researcher, and operational practice is in its turn composed of a practical problem which is performed for an operation’s account. The person who carries out action research is sitting on two chairs; in the practical work one is acting a consultant and in the scientific work one is acting researcher. The two practices also generate two types of knowledge, scientific contributions from research and practical knowledge from the operational practice. The figure below describes how action research fit together and describes how they have been used in this case.
Figure 2. The meeting between the two types of design research.

It is not necessary that the two research practices are performed by two persons, in this case they have been carried out in parallel and the same person has occupied both roles. Instead of a researcher studying the consultant’s work, the researcher has studied herself. By applying the methods regarding usability and evaluate them a continuous research work was carried out in parallel to the operational problem with the prototype.

The practical work has followed an iterative process, where the phases of analysis, design, prototype construction and evaluation are repeated until the interface has reached its final structure and the result is accepted by the involved parties (see Figure 3). The theory from literature studies and the empirical data that emerged from the interviews and the other methods have formed the basis for further work on the development of a new interface for the application.

Work began with a Heuristic Evaluation to evaluate the existing interface. After that interviews were made with responsible persons at Ericsson to get an understanding of what the common opinion about functionality and weakness in the current implementation was. The reason to the selection of methods was to get both the interaction designer’s, the developer and the testers views of the system. If a problem was detected in both the Heuristic Evaluation and group interviews it was considered an important problem, and something that should be reviewed. The problems detected in the Heuristic Evaluation were used to define questions to the experts. Both the Heuristic Evaluation and user tests focus on the weaknesses, not the strengths, of the system. The result may therefore indicate a great dissatisfaction with the system because the strengths have not been mentioned.
4.2.2 Observation of method application

Action research has been applied in the investigation since the double processes of design research has been used. In many cases a researcher is studying an operation and then writing a thesis on it. In this case the researcher has studied herself. A double role has been required to perform evaluations that the interaction designer has analyzed herself. I have recorded the Focus Groups with handheld computer to be able analyze the data afterwards and made notes during the process.

4.2.3 Method analysis

The methods were then evaluated to obtain each method's strengths and weaknesses. Then the methods were compared from three aspects which were time, resources and results.

4.2.4 Implementation of Heuristic Evaluation

The Heuristic Evaluation was performed first, to catch as many usability problems as possible at an early stage and to get a deeper knowledge about the application’s scope and interaction at the same time. The evaluation was carried out by the interaction designer and was divided into several occasions, since the application was quite difficult to understand. As a new user an introduction was needed first and foremost, to understand the purpose and functions of the application. Time was spent to explore
and familiarize myself with the navigation and the existing features before the evaluation could be started. The approach was to initially identify the shortcomings of the system and then find what guideline each deficiency violated. Deficiencies were discovered by performing the same tasks as a user performs when using the application, and when a question or ambiguity arose this was registered. The method was applied to the existing application design and was repeated several times by exploring the structure, functions, menus, buttons and so forth. The results were compiled in a table. Finally each usability problem was given a grade which acted as a priority scale based on how serious they were and which problems that needed to be addressed first. This is the scale that was used:

0  No problem at all / will not be possible to assess  
1  Cosmetic problem, dealt with if time permits  
2  Small usability problem, should be addressed with low priority  
3  Big usability problem, should be addressed with high priority  
4  Very high usability problems, must be addressed before the product is can be regarded as finished.

4.2.5 Implementation of the Focus Groups

In order to identify the shortcomings of the application and to obtain usability requirements Focus Group interviews were held involving people at Ericsson. The interaction designer represented the role as moderator and supervised the meetings. The importance of having developers and testers, who are experts on the application, involved in the design process was of great significance. They are very familiar with the application and some of them had met users. They constitute a source of knowledge that is the result of many years of developing the product. A relative unstructured form of group interview is best suited when the purpose is to create a free discussion by brainstorming. The Focus Group interview had five participants; four programmers and one tester. There should be a moderate number as Victoria Wibeck writes there should not be fewer than four and not more than six people in a group interview (Wibeck, 2000). In order to recruit members an email was sent with the invitation to the development team. As the selected participants work together in the same team and were good colleagues, they represented a homogeneous group. The positive side of a homogeneous group is that nobody is afraid of entering the discussion. There is a risk that certain subjects are not brought up since there are
things that are taken for granted within the group. Everyone knows what the others think on certain issues and therefore it might feel unnecessary to bring them up. A homogeneous group provides intimacy and consensus among the participants. To be able to re-listen to the material and analyze it several times, the group interviews were recorded. All problems in the application that were revealed during meetings will be communicated through a new design proposal. It will be described in the Results chapter how each problem has been resolved.

**Focus meeting 1:** 5 participants + moderator

**Focus meeting 2:** 5 participants + moderator

**Focus meeting 3:** 7 participants + moderator

**Focus meeting 4:** 7 participants + moderator

### 4.2.6 Implementation of Cognitive Walkthrough

Cognitive Walkthrough is a simulation of the user’s problem-solving process where a prototype is used to evaluate the steps in the tasks that the user will perform. The evaluation is conducted by the designer and several expert evaluators and is performed on the basis of the new prototype to detect defects. Each task is performed stepwise and in the meantime the following questions are answered:

- Does the user know how to complete the task?
- Will the user detect that the correct action is available (in the form of buttons and menus)?
- Will the user understand that they have done the right thing from the available feedback?

The method was carried out partly in a group consisting of developers and the interaction designer. It was supplemented by individual work of the interaction designer. The method used when updates to the design had been made. Beside the developer’s own comments focus were put on the tasks of the users and if something was lacking from that aspect. During the survey critical information and notes were made about the design changes that need to be implemented to solve the problems that had been encountered.
4.2.7 Implementation of Action Analysis

The purpose of Action Analysis is to predict the time it takes for a user to carry out a task. The method was performed on both the new and the existing application to see the differences, and if the improvements in work efficiency and ease of use were as expected. It was not possible to measure the time precisely since the prototype was not fully working. Usually the measured steps in an Action Analysis are so small that they only take a couple of seconds, or even fractions of a second, to perform. But to measure with such detail special software is required, which was not available for this work. Instead larger tasks were chosen that could be timed manually with a clock.

The tasks examined were the following:

1. Find the counter “Drop IMEI” by using the search function.
2. Add 50 copies of this counter to two different counter groups.
3. Set one filter to the same value for all these counters.
4. Rename 5 of these counters to Adam, Bertil, Caesar, David and Erik.
5. Move 20 counters from the first counter group to the second counter group.

4.2.8 Prototype construction

A prototype is a test version of a product to be able to interactively test the interface design. After the methods thoroughly had been performed and the problems had been revealed, the purpose and needs of the product were identified. Each problem should be solved to make the application more user-friendly. The sketching of the interface could then begin, and a rough sketch was made to structure the ideas that had formed during the execution of methods. After this the work began in Netbeans, which provides complete GUI components that can be used as building blocks for creating an interface. GUI components are the components that build an interface, from windows to buttons, menus, check boxes and drop-down menus. The program proved to be handy to work with and was more efficient than expected. Since Netbeans worked so well there was not a need for many hand-drawn sketches. The prototype visualized the design ideas in a more thorough and time-efficient manner. The ideas were tested with iterative prototype development; the more iterations the better result.
5. Results

This part describes the results of previously described methods and results on the new design proposal.

5.1 Results of Heuristic Evaluation

The table below lists all deficiencies that the Heuristic Evaluation found in the application that violates Jakob Nielsen’s guidelines. These deficiencies have been ranked on a scale from 1 to 5, to gain an overview of what should be prioritized during the design. 0 is least severe and 5 is a very serious problem. The foremost priority issues were considered obscurity, lack of help functions.

Table 1. Encountered deficiencies in the application, and what guidelines they violate.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Violates heur. no.</th>
<th>Severity (1-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The menus contain no shortcuts</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>2. Cumbersome to select filters</td>
<td>2, 7</td>
<td>4</td>
</tr>
<tr>
<td>3. No good alternative to drag-and-drop</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>4. Unclear search function</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5. There is no undo button</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6. Drop-down menus with only one option</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>7. Difficult to distinguish between different counters</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>8. The total number of counters is not visible</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>9. Copy function missing</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10. Many mouse clicks</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>11. Lack of help functions</td>
<td>9, 10</td>
<td>4</td>
</tr>
</tbody>
</table>

Below is a detailed description of identified usability shortcomings.

1. The menus were first studied and it was discovered that they had no shortcuts. It is still possible to use the menus from the keyboard by pressing a mnemonic, Alt+menu letter (Cooper, 2007), but shortcuts can make the interface more efficient to work with. This violation received the lowest number in the priority scale since it was considered only a small problem. Shortcoming violates Guideline 7, “Flexibility and efficiency in use”.

2. Selecting filters for a counter was a cumbersome process since the filter settings for a counter only are visible if the counter is selected. The counters had to be
selected one by one. It was not possible to select several counters at the same time if the same filter was to be changed for all these counters. This was considered to be a major usability problem because it is a main task in the application. The weakness was considered to violate guideline 2, “Match between system and real world” and 7, “Flexibility and efficiency in use”.

3. For a new user it is difficult to find so-called “hidden” features such as drag & drop. The function was not found in the menus and in the beginning it is tricky to understand how to add a counter. The alternative to drag & drop was to double-click, which is also a hidden feature that is not easy to detect. The shortcoming was in violation of guideline 6, “Recognition rather than recall”.

4. The search function is not user-friendly as the matches ended up in an expanded tree structure. The search matches were selected in the tree structure one at a time but the counters that did not match the search string were not hidden, so you might have to look for the match. The shortcoming was in violation of guideline 6, “Recognition rather than recall”.

5. There were no established commands such as Ctrl+Z to undo an action which caused trouble if the user added the wrong counter by mistake. The shortcoming was in violation of guideline 3, “User Control and Freedom”.

6. In the filter settings frame there were drop-down menus with only one option, which required unnecessary clicks when the sole option could have been selected by default. The shortcoming was in violation of guideline 7, “Flexibility and efficiency in use”.

7. There were counters that were marked with (#) and (%) depending on what the represented. These should easily be distinguished in order to create better clarity. The weakness was in violation of Guideline 6, “Recognition rather than recall”.

8. Counter definitions can, as mentioned before, tend to become very large and then it is a plus if you quickly can see how many counters the definition contains. The shortcoming was in violation of guideline 1, “Visibility of system status”.

9. To copy items with Ctrl+C is a well-known command that was not available in the application. For instance, if the user wants to create more instances of a counter it was not possible to copy and paste it; the user had to drag & drop it and set the filters for it again in order to create a copy. The shortcoming was in violation of guideline 3, “User Control and Freedom”.

23
10. The application required many mouse clicks, especially when each counter had to be modified separately. The shortcoming was in violation of guideline 7, “Flexibility and efficiency in use”.

11. The online help texts in the application were not sufficient and certainly not for a new user. It was not possible to know what a specific counter counted, without downloading and reading a document from the documentation library home page. This shortcoming was in violation of guideline 9, “Help users recognize, diagnose and recover from errors” and 10 “Help and documentation”.

5.2 Results of Focus Groups

The first Focus Group interview resulted in improvement suggestions from the participants. The interview could be summarized in a short list of the most problematic shortcomings in the existing design:

- Poor overview
- Poor search function
- No copy support
- A slow way to create and change counters
- No information about the counters

The subsequent Focus Group interviews resulted in feedback on the updated design proposals, where the participants could share constructive criticism. The ideas that were considered to be possible to use were implemented.

5.3 Results of Cognitive Walkthrough

The following tasks were gone through:
1. How does the user find the right counter?
2. How does the user add a counter?
3. How does the user add more than one counter at the same time?
4. Will the user understand the distribution dialog?
5. Does the user understand what the filter value “-“ means?
Results:

1. The user discovers that tooltips are available when the mouse hovers over the counter. The user wants to find a specific counter and now starts looking for the search function. There is no Ctrl+F shortcut, which is available in most other applications, but the search tab is always visible in the interface and should be very easy to find anyway. The search controls (the text field, the Next, Previous and Show All buttons) are easy to understand. The drop-down menu with the search area might take a while to get acquainted to, but the text descriptions are self-explanatory. The user thinks that the system gives clear feedback since only the search results for the written search patterns are shown and all other counters are hidden. The user feels confident in the system since the functions are visible at several places in the interface.

2. The user guesses that the toolbar button with a “+” icon could be used to add counters. He hovers the mouse pointer over the button and gets a confirmation when the tooltip text says “Add”. He uses the toolbar button to add a counter. He later finds out that it also is possible to right-click and chooses “Add” from the popup menu, that the Add option is available in the window menu bar and that it is possible to drag & drop a counter too, but these options are not as visible as the toolbar button. Although he used the toolbar button the first time he will use keyboard shortcut to add counters from now on, since it is more efficient than clicking a button.

3. The user would now like to add a variety of counters, and tries to make a "multi-selection", that is select multiple counters by pressing the Ctrl + mouse click on the counters he wants to add. After the desired counters have been selected he right-clicks to bring up the popup menu, which most Windows users are accustomed to. He notices that the popup menu looks the same as when only one counter was selected and clicks “Add” to add the counters. In the future he will use the keyboard shortcut to add several counters since it is more efficient.

4. The user is now curious of what the toolbar button next to the “Add” button does. The icon also contains a “+”, but also a folder, so the user assumes that this is a more complex way to add counters. He selects a counter and clicks the button. This brings up a dialog on the screen that explains that this option will create one new counter for each selected filter value combination. Since this text describes the function there is no need to read the online help text. The user finds out this options also is available in the menu bar, in the popup menu and that there is a keyboard shortcut for it.

5. The user has now selected counters he does not want to have any filter for. The default filter value is “-“ so he assumes that this means that no filter is active, but to be
sure he opens the online help text by pressing F1, as in most Windows applications. The help text describes how a filter value is set and confirms that “..” means no filter value.

5.4 Results of Action Analysis

Table 2. Times to perform the tasks in the old and new design.

<table>
<thead>
<tr>
<th></th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>2</td>
<td>32 sec</td>
<td>13 sec</td>
</tr>
<tr>
<td>3</td>
<td>175 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>4</td>
<td>20 sec</td>
<td>20 sec</td>
</tr>
<tr>
<td>5</td>
<td>81 sec</td>
<td>5 sec</td>
</tr>
</tbody>
</table>

“Old” means the current application and “new” is the new prototype. The tasks were chosen to resemble of common user operations and the tasks were performed in the two design proposals to see if the new design contributed improvements.

The data examined were the following:

1. Find the counter “Drop IMEI” by using the search function.
2. Add 50 copies of this counter to two different counter groups.
3. Set one filter to the same value for all these counters.
4. Rename 5 of these counters to Adam, Bertil, Caesar, David and Erik.
5. Move 20 counters from the first counter group to the second counter group.

How each task was performed:

1. **Old:** Wrote the search name in the search field and pressed Enter. Jumped to the next search match by pressing Enter again, until the counter was found.
   
   **New:** Wrote the search name in the search field and pressed Enter. Scanned through the list of matches to find the counter.

2. **Old:** Added 25 counters to the first counter group by double-clicking. Added a new counter group and added 25 more counters to it in the same way.

   **New:** Added a counter, select it and pressed Ctrl+C. Pasted this counter four times with Ctrl+V. Then selected all five counters and copied and pasted them four times to create a total of 25 counters. After this a new counter group was
created. Copied all 25 counters from the first counter group and pasted them into the second.

3. **Old:** Selected one counter at a time and then selected the filter value for each of them.

   **New:** Set filter value for a counter. Selected the filter cells for the other 24 counters in the group and pasted the value. Selected after that the filter cells of the 25 counters in the second group and pasted the filter value.

4. **Old:** Selected each counter separately and renamed them in the name text field.

   **New:** Select the cell with the column name “Counter names” and entered the new names in each cell.

5. **Old:** Deleted 20 counters in the counter group and then added 20 new counters to the second counter group by double clicking it 20 times. Since the filter values were lost these had to be set again for the 20 new counters.

   **New:** Selected 20 counters in the first counter group, pressed Ctrl+X and then pasted them to the second counter group with Ctrl+V.

### 5.5 The new design proposal

This chapter begins with an explanation of the design in general and then it goes into more detail and describes the system’s identified problems and how the new design solves them. Each statement is also shown in pictures, where the differences in design are clarified.

#### 5.5.1 Application of different frames

The new design (Figure 4) now consists of a menu bar, a toolbar, search bar and three frames. The biggest frame is the one that the user mainly is working in, which is where the counter data and filters are edited. The two smaller frames to the left of the window are overview frames; one with a tree structure that contains all available counters and one with a list of all created counter groups. The font used for all text in the interface is the same as Windows uses; the prototype reads the font setting in Windows and uses the same font since this is the default behaviour in Java GUIs. Color choices are neutral and the reason for having two different colors for the table rows is to make it easier to distinguish them so the user does not edit a table cell for
the wrong line by mistake. The light yellow color is \((R, G, B) = (240, 235, 210)\) and the
dark yellow color is \((R, G, B) = (230, 215, 145)\). All icons used are taken from the
Eclipse application. The frames are used to encapsulate different functions into
different areas. The tabs (frames) are dockable to allow the user to create a personal
layout, move and resize them for an efficient way of working. The search function has
been developed by making both the frame with available counters and the frame with
selected counters searchable.

Figure 4. Display window of the new design proposal, EBS.

1. **Menu bar.** To the menu bar (Figure 5) a menu has been added called “Counter
Group” which can be used to create new counter groups and to compare an edited
definition to the last version that was activated by “apply”. One can also undo all
changes made to a counter group since it was with the menu option “Revert to applied
version”, and switch between two filters views; expanded or collapsed view. The
collapsed view packs all filters into one column each from left to right. The column
“Filter 1” contains the first filter, “Filter 2” the second filter and so forth. In the
expanded view the filters are spread out so that each filter gets its own column.
For instance the filter Cause Code would get its own column. This column would be
empty for counters that do not have a Cause Code filter. This means that the table will
become very wide and that most table cells will be empty, but it allows the user to
quickly find all selected counters that have a certain filter type. The columns are
sortable; if the user for instance clicks the Cause Code column header all counters that have this filter type will end up at the top, and all counters that lack this counter type will end up at the bottom.

Under the “File” menu a print function has been added that makes it possible to print counter definitions to paper. Under “Edit” a number of options have appeared: Undo, redo, cut, copy, paste and delete. Under the menu heading “Window” it is possible to freeze the layout so the user does not move a frame by accident and changes a layout that one has configured.

2. Toolbar. A toolbar was added to highlight the most important functions and provide the user with the functions he needs quick access to. Menus are indeed a complete tool set to learn the application but the toolbars make the most commonly used commands more accessible. For most toolbar options there are shortcuts to speed things up for more experienced users.

3. Available counters. The tree structure has been kept since it gave a good overview of the available counters. Various types of counters can be grouped into folders and folders can be expanded to produce a list of the content.

4. Selected counters and counter groups. A table, which has the purpose to function much like a spreadsheet in Microsoft Excel, has been added to create a more efficient work process for the user. For each added counter a new row is created in the table. The rows can be copied, moved and deleted. Filter values are selected by double-clicking a filter cell. A dialog will pop up with a filter value editor where the filter value is selected in different ways, depending on if it is an integer, free text or pre-selected alternatives. The various alternatives to select filter values are the following:

**Integer:** Spin button with a min and max limit (Figure 6).

**Free text:** Mainly for intervals, for example “1-4, 2-8” (Figure 7).

**Drop down list:** For pre-selected values where only one value can be selected (Figure 8).

**Multi-select list:** For pre-selected filter values where one or more options can be selected (Figure 9).
Figure 6. Spin button for integers.

Figure 7. Free text for interval values.

Figure 8. Dropdown list for single-selection.

Figure 9. Multi-select list.
5. Overview of the counter groups. The purpose of the frame is to get a quick overview of the counter groups, and how many counters each counting group contains.

6. Search field. Search field acts as a traditional search in, for example, Windows Vista, it is possible to look at the sub-string if you do not remember the full name of what you are looking for or if you just want to streamline the search. The search functionality has been expanded through various search combinations where it is possible to search in two different frames and where you can jump between different matches, or alternatively view all at once. Detailed description on the search is later in the results.

5.5.2 Poor overview

Previous solution: The selected counters ended up in a tree structure (figure 10), where each counter had to be modified individually, which was time consuming and required many mouse clicks. There was no opportunity to see the total number of counters that had been added to the definition. It took a very long time to count them manually as a definition can contain thousands of counters.

![Figure 10. It was not possible to see how many counter the definition contained, or the filters that had been configured for the counters.](image)

New solution: The frame that contains the selected counters has been changed to a dynamic table where the selected counters may be modified, copied, moved or deleted. The rows are numbered and each counter occupies one row in the table. Filter values can be directly connected to their corresponding counter, and the same filter value can...
be applied to several counters at the same time. The former counter groups which
where symbolized with folders will now appear in the form of tabs, each tab
represents a counter group (figure 11). The idea is that the table will serve as a
worksheet in Excel. It should not be possible to write or paste non-valid filter values.
A toolbar has been added to make the application more efficient and to give the user a
better overview.

**Figure 11.** There is one tab for each counter group.

To get an overview of the number of counter groups that had been added, a counter
group frame has been introduced (figure 12). Here, the user can see how many
counters each group contains and easily delete whole groups and rename them. It is
also possible to double-click a counter group to open its counter group table frame.

**Figure 12.** Overview of counting groups containing counters.

Features have been visible in several places to increase intuitivity and increase the
user's freedom of movement (Shneiderman, 1995). Dynamic and movable tabs have
given the possibility for users to create their own layout and organize the tabs for their
convenience. Tabs are moved by dragging and dropping them and it is possible to
distinguish between them as they pulled out to the desktop and open in a new
window. Features can be activated through the menu and tools menu (see Figure 13)
and the functions' add 'and' distributor 'is possible by right-clicking on the calculator.
Figure 13. Toolbar menu and its buttons.

A toggle button has been added in the toolbar to make it possible to show the frame of selected counters in two different views. A toggle button is used in interfaces to enable or disable an option, just like a switch is used to turn on or off a lamp in real life.

5.5.3 Poor search function

Previous solution: The search function only searched for counters among the available counters, that is, in the left frame. The first match was selected, and by pressing Enter the next match could be selected (figure 14).

New solution: The new solution has a search tab that allows searching both among available and selected counters (figures 15 and 16). One specifies the search area in a
drop-down menu, and you can choose to only search among the available counters, only among the selected counters, or in both tabs simultaneously.

When you search among available counters, the counters and folders that contain the specified substring will be shown while the others are hidden, which makes it possible for the user to quickly find the counter he is looking for. If the user, for example, writes “DC” the counter named “SDCCH Drop” will be shown while the counter “RX Qual DL” will be hidden.

When searching in the table of available counters marked first table cell containing the search string. You can skip to the next or previous result with the Next and Previous buttons. It is also possible to highlight all matches at once with the Show All button. In the drop-down menu with search area you can also choose if you just search in the selected group expects tab, or skip to the next group when counting reached the end of the first.

![Figure 15. The search function can search in the application's two frames.](image)

**Figure 15.** The search function can search in the application’s two frames.

![Figure 16. The search area settings.](image)

**Figure 16.** The search area settings.

### 5.5.4 No copy function

**Previous solution:** This included, as previously mentioned, that the application did not provide copy function where filter values could be copied to several counters simultaneously. For example, if the counter "SDCCH Drop" would be used several times the user had to do drag & drop it as many times as the desired number of counters, instead of creating one counter and then copy it (figure 17).
Figure 17. *The application did not provide any copy function.*

**New solution:** It is now possible to copy the cell values or entire rows in the counter table. This makes it easy to move a counter to another group (select the row, cut and paste it in another counter group tab). The procedure can move a counter to another location within the same group.
But the biggest benefit is that it is easy copy filter values. A cell can be selected and copied, and then the value can be pasted into multiple cells in the same way as in Excel. Assume that the user has 30 counters with the same filters. Now he wants to change the filter values of all these counters. He copies the cell with the first filter value and then pastes it into the remaining 29 filter value cells for the other counters. This feature allows the user to paste filter values that are not allowed for a counter, and it must be controlled by the application so that only valid values can be pasted.

Figure 18. **Cut, copy and paste increases the application’s efficiency.**

### 5.5.5 A slow way to create new counters and to change them

**Previous solution:** To create a counter it had to be dragged & dropped to the tree.

**New solution:** Now filter values can be changed in a quicker way, because the user does not have to first select the counter to get access to its filter settings. In the new solution a filter value is changed by double-clicking a table cell. Moreover, the Distribution function has been improved. It creates one new counter row for each
selected filter value combination. The filter values can then be edited, which was not possible in the previous solution.

5.5.6 No information about counters

**Previous solution:** The online help contains no information about the counters. To get information about them the user had to download the “Monitor Description” document from the documentation home page. The help menu (figure 19) contained a lot of options, but they were confusing since the purpose of some of the options was unclear.

![Help Menu](image)

**Figure 19.** The help menu.

**New solution:** In the new design a brief help text is displayed in the form of a tooltip then the mouse pointer hovers over a counter (figure 20). This tooltip is available both in the available counters tree and in the counter group tables. Furthermore, you can select a counter in the tree or in the table and press F1 to open the online help page for this counter.

The “help” menu has been modified to make it more standardized (figure 21). It has been reduced to only contain the “Content”, “Index” and “Find” found in most other applications.
5.5.7 Proposals that were not included in the new design

The text below describes some of the proposals from the Focus Groups that were not included in the prototype, for different reasons:

**Context-sensitive search:** This meant that the search would search only in the area where the user last worked. This proposal was not chosen because it can be unclear what the last activated context is. An example would be if the user selected a counter and dragged & dropped it from the available counters tree to the selected counters table. You might then want to search for a new counter among the available counters but the context-sensitive search function would instead search in the selected counter table since it was the last activated context.

**Sandbox:** A proposal that was expressed was to create a "sandbox" where users could create a temporary statistical definition. The proposal was not included since a "dummy" counter group can be created that the user never activates.

**Undo function for each counter table cell:** The proposal was to make it possible to select table cells and undo the last change to them only. This would be very tricky to implement.

**Show possible filters for each counter:** The proposal was to make it add a menu
item “Show all possible filters” to the popup menu when right-clicking a counter in the available counter tree. The value of this feature was considered low, since all the filters clearly will be shown when the counter has been added.
6. Discussion

This chapter discusses the work; evaluation methods are compared with each other and examined with critical eyes. The thesis ends with a conclusion and suggestions for further work are presented.

6.1 Performance Problems

Table 3. Matrix with compilation of the resources, time and results of the used methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recourses (in terms of people)</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus Groups</td>
<td>participants + moderator</td>
<td>7 h</td>
<td>Discussed under heading 6.2.2</td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>evaluator</td>
<td>16 h</td>
<td>Discussed under heading 6.2.3</td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>evaluator</td>
<td>8 h</td>
<td>Discussed under heading 6.2.4</td>
</tr>
<tr>
<td>Action Analysis</td>
<td>evaluator</td>
<td>8 h</td>
<td>Discussed under heading 6.2.5</td>
</tr>
</tbody>
</table>

The comparison shows that it did not take a great deal of resources in terms of people to perform any of the methods. The most resource demanding method was Focus Groups, where 5 – 7 people + a moderator were involved. Most time was spent on Heuristic Evaluation, and it was also the one I thought found the most usability problems in the evaluation. Results of the methods are discussed later in this chapter.

I think that usability tests are very important and that more companies should invest in this to correct the problems before the applications are delivered. Since there is fierce competition today on the market and the customers might choose another supplier if the application does not meet the usability requirements. A combination of evaluation of a design and coding by developers is a good way to create an interface, according to my opinion. The interface design team can then focus on the usability problems while the developers focus on the coding instead of both parts.

I believe that it is not possible to get away from that the best way to get good design ideas is to make usability studies and then base the design on the information and the users’ reactions. User involvement in the development is particularly important when systems or websites are developed for a large user base such as news portals or search portals. In this case the target group was limited to mobile operators, so if more information about them had been available it is most likely that the methods would have been more successful. For instance, if the end user’s technical conditions would
have been more known it would have been possible to optimize technology and function for maximal usability. Working iteratively in the design process is more or less a must. For me this way of working worked very well. As the designer one cannot sit in a corner for ten weeks and then present the proposal, it must be presented to the stakeholders on a regular basis and it has to be checked that no major misunderstandings have been made.

6.2 Methods

Broadly speaking the procedure of applying methods has worked according to plans. By using audit-based evaluations, it was possible to find potential usability problems at an early stage. It was interesting to see how the different methods actually resulted in different types of usability problems. It turned out that the Heuristic Evaluation reflected many specific types of problems while the other methods highlighted problems that were related to the proposed user’s mindset. The disadvantage of all of the methods was that they will find fewer problems than user tests. In addition, auditors and users value the problem in different ways. The Cognitive Walkthrough found 40-50% of the problems. The methods only found potential problems in the interface while tests with real users discover real errors. One positive side of the implemented methods was that it was possible to implement them both individually and in groups. The main advantage of the methods was that they revealed shortcomings in the interface at an early stage and that they did not require a lot of resources; time did not have to be spent on contacting users, money did not have to be spent on travel tickets and so forth. It has to be noted that there might be problems with the results of all these methods since they often are based on the opinion of just one person. The most rewarding way to evaluate an interface design would have been, according to my opinion, a combination of examination-based evaluations and user tests. Nielsen has shown that errors can be found more quickly if the evaluators are experts in the domain or in user interaction, or even better; in both (Doubleday, Ryan, Springett, Sutcliffe, 1997).
6.2.1 Critical examination of Focus Groups

Focus Groups were a good method as these contributed with the primary data to the work, and it was from this information that the new interface design emerged. Acting moderator was not a heavy task; since the group had strong and interesting opinions and thoughts on design all the meetings were smooth runs. The time to speak was fairly spread within the group, but the moderator had to break off and start a new aspect of the subject from time to time. The Focus Groups method was chosen because:

- Group interviews were more appropriate than individual interviews, since they allow a broader scale of ideas to be expressed.
- The purpose was to listen to what the group members felt were important aspects of the substance, not to have a number of questions answered
- Time-saving method

Recording all Focus Groups also eye contact and interaction between members easier, as the moderator did not have to take notes during the meeting. At first the members questioned the recording and it was noticed that some members felt a little uncomfortable, but they seemed to have forgotten the recording after a while. Instead, the moderator could be involved in the discussion. Comments on the new design proposal were organized through Focus Groups and the number of participants was limited, as the method does not allow for many group participants. Ericsson's employees had very good ideas, and I chose to implement several of them, but some were rejected, partly because it is not possible to meet all demands and partly because I could use my knowledge to argue why some of them should not be implemented.

6.2.2 Critical examination of Heuristic Evaluation

Nielsen claims that this method is fast, easy and very cheap (Nielsen, 1999). The advantages are that the resulting material is structured as it is based on guidelines and as they are analyzed in order, one by one. The disadvantage of the method, according to Nielsen, is that it is difficult for one person to find all usability problems in a user interface. In addition, auditors and users value different problems in a different way. Based on experience from several projects Nielsen could note that a single assessor only found 35% of the existing usability problems, and that five evaluators found 75% of the problems (Nielsen, 1999).
On the positive side the method a range of errors, from smaller ones to more serious errors. But it feels as if there is a risk than “false alarms” are created. I think it was an efficient and fruitful method to identify problems in the interface. Nielsen believes that errors are found more rapidly if the evaluators are experts in the domain or in user interaction. I think of course that this is true; if the evaluator possesses knowledge in both system and user-friendliness there is a big chance that the evaluation will be really good. Nielsen says that the evaluator of a Heuristic Evaluation determines exactly how to proceed, but Nielsen recommends that the interface is gone through at least twice. The first time is to get an idea of the interaction flow and the general architecture of the system, the second time is to focus on specific aspects of interaction (Doubleday, Ryan, Springett, Sutcliffe, 1997). This depends on how sophisticated the interface is but I think that time should be spent on the Heuristic Evaluation to find all problems, so I think that two times is absolute minimum. A minus with the method was that problems were identified without any suggestions on how to solve the problems. The result was only a list of deficiencies, and then it was up to the evaluator to find solutions. The fact that the method takes no account to how the actual end users think and feel is a disadvantage. Despite the drawbacks I felt that the Heuristic Evaluation was a relatively simple and cost efficient way to make sure that at least some of all the design guidelines are followed.

6.2.3 Critical examination of Cognitive Walkthrough

This method had a limitation because it only targets how easy an interface is to learn. However, I think that this can be an advantage, to learn a system easily and quickly is very important, especially for new users. An interface should be as self-teaching as possible (learning by doing), so users would not need to read the manual. A general disadvantage of the method was that it required a detailed prototype to carry out this review, but in my case it fit well since the goal was a prototype with as much detail as possible. It felt like more information was needed about the users or how new users tend to think and experience a new system. The implementation of the method tended to feel fictional, since I did not know how a new user would interact. This is mainly because I had no opportunity to consider the knowledge of the users.

I also had relatively high expectations about this approach and that it would be fruitful. Lewis and Riemann (1994) argue that the most successful walkthroughs are made by designers who have worked close to real users, so they can create a mental
picture of them and their current environment. After carrying out the method, I can agree with this. The result of the method became a story how a user step by step performed a task, and this story contains risks for the user to make errors. The documentation can then be used to correct the design and reduce the risk that the mistakes are made. The method also gave a good opportunity to think through the functions and the design solutions in detail, but despite this I believe that the method does not fit perfectly in my case since I did not know enough about the users.

6.2.4 Critical examination of Action Analysis

Action Analysis was a hilarious method to perform there I worked as a user by performing tasks in the two applications. By analyzing the different steps one hand required to perform a task attempts to locate the problem. Examples of problems might be that it required too many steps to a given task. When I carried out the data I got to feel what it was like to work in applications and the differences in time became quite obvious. Especially the features cut and paste and that it is possibility to see how many counters you have in a definition eased significantly in these executions. With the results to judge, I believe that the new design can be classed as more efficient to work in. It does not differ much in time on all tasks but there were relatively large differences in the other.

6.3 Conclusion

My research questions were: What methods are appropriate to use when you do not have access to users in a design process? What methods to evaluate the usability of the interface proved to be most successful and what were the benefits and disadvantages of each method?

The method with most advantages was Heuristic Evaluation since the design is compared with well established and well proven guidelines. The disadvantage is that it takes a long time to perform. Cognitive Walkthrough is not worthwhile if there is not so much information about the users. Then it becomes more of a game of guesses. The method is more useful if it is applied at an early stage and if there is detailed information about how the users interact with the system. Focus Groups turned out to be very useful if the participants, as in this work, had a lot of experience and knowledge about the subject. Action Analysis requires special software in order to measure with the fraction-of-a-second accuracy which actually is needed. It provides
hard facts in form of measurements that the new design is an improvement and it is also efficient to find bottlenecks in an existing interface. Common for all methods is that the result is more reliable if more people than one act as evaluators. To summarize the methods found shortcomings that really have improved the interface.

6.4 Suggestions for further work

Further work in this area could be to implement the prototype in Java and integrate it in the real EBS application. The visualized prototype and this report, that describes all its functions, will serve as a basis. Furthermore all parts of the EBA applications could get a similar look and feel. They do not have to be completely identical, but docking windows and a toolbar where all commands easily are accessible would be good to implement.

In the context for this study only a certain analysis has been made regarding user-friendliness. There are more methods to evaluate interfaces and they could also have been applied. In that case there are no guarantees that more shortcomings in the interface would be discovered.
7. References

7.1 Books


7.2 Articles


7.3 Electronic sources


8. Attachments

The first hand-made sketch before the design was done in Netbeans.

8.1 Attachment 1: <Design sketch 1>