Usability in Information Handling Applications

An Evaluation and the Development of Design Guidelines for the Web Application Docomotive® at Sigma Information Design

RICKARD ANDERSSON, ANNA PERSSON

Department of Computer Science
IT UNIVERSITY OF GÖTEBORG
GÖTEBORG UNIVERSITY AND CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2006
Usability in Information Handling Applications
An Evaluation and the Development of Design Guidelines for the Web Application Docomotive® at Sigma Information Design
RICKARD ANDERSSON, ANNA PERSSON

© RICKARD ANDERSSON, ANNA PERSSON, 2006.

Report no 2006:22
ISSN: 1651-4769
Department of Computer Science
IT University of Göteborg
Göteborg University and Chalmers University of Technology
P O Box 8718
SE – 402 75 Göteborg
Sweden
Telephone + 46 (0)31-772 4895

Tryckeriet, Matematiskt Centrum
Göteborg, Sweden 2006
SUMMARY
Software applications under continuous change and improvements are often developed by several developers with different skills, background and knowledge. This can sometimes be seen in the graphical user interface regarding layout, style and design concept with lack of usability issues as a result. The web based application Docomotive® at Sigma Information Design was in need of a usability evaluation to improve usability and conformity. Surveys, interviews, heuristic evaluation and observations with think aloud technique were methods used to gather usage data. The data was analyzed and the result showed that the graphic user interface of Docomotive suffered from several usability problems and improvements were needed. Since Docomotive is constantly adapted to fit new project needs, a usability design guideline was produced for developers to follow. Crucial usability problems were implemented to serve as examples for the rest of the application and a list of usability problems to be corrected was delivered.

The report is written in English.

Keywords: Web application, document information, usability analysis, Graphical User Interface, design proposals, design guidelines
ACKNOWLEDGEMENT

This Master Thesis is the final work at the M. Sc. Program in Human Computer Interaction/Interaction design at the University of Information Technology, part of Chalmers University of Technology. The thesis consists of 20 credits which is equivalent to 20 weeks of work. The thesis and work was carried out at Sigma Information Design in Gothenburg.

We want to thank our corporate supervisor, Johan Thornadtsson, for his support and guidance as well as the coworkers involved at SID for being helpful and cooperative. We would also like to thank our supervisor Olof Torgersson at the IT University.

Rickard & Anna

Göteborg, March 6th 2006
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.1 Background</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.1.1 The Company</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.1.2 The Problem</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.1.3 The Task</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.1.4 System description</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>LITERATURE</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1 Human Computer Interaction</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1.1 Diversity of Human Computer Interaction (HCI)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1.2 Goals with Usability in User-Interface Design</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1.3 Benefits and Potential Outcome with Usability Engineering</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2 Interaction Design</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.1 The Interaction Designers Role in the Design Process</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.2 Methods and Tools</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2.2.3 User-Centered Design (UCD)</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2.3 Cognitive Perception</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2.3.1 Color</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2.3.2 Icons</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>METHOD</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>3.1 Planning the Process</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>3.2 Identifying the Users</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>3.3 Choosing Methods for the Usability Analysis</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>3.3.1 Feasibility study; Context of use, Interviews and Literature study</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3.3.2 Survey</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>3.3.3 Heuristic Evaluation</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>3.3.4 Observation (Think-Aloud Technique) with complementary interviews</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>3.4 Conducting the Usability Methods</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>3.4.1 Feasibility Study</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>3.4.2 Survey</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>3.4.3 Heuristic Evaluation</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>3.4.4 Observation / think aloud method with complementary interviews</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>3.4.5 Telephone Interviews</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>RESULT</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>4.1 Gathered data</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>4.1.1 Survey</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>4.1.2 Heuristic Evaluation</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>4.1.3 Observations (with Think-Aloud Technique)</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>4.1.4 Telephone Interviews</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>4.2 Analysis</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>4.2.1 Navigation</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>4.2.2 Feedback</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>4.2.3 Language use</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>4.2.4 Icons</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>4.2.5 Colors</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>DESIGN PROPOSALS</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>5.1 Proposed Design Solutions</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>5.1.1 Areas, proposal 1</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>5.1.2 Areas, proposal 2</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>5.1.3 Search, Proposal 1</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>5.1.4 Edit Document Information, Proposal 1</td>
<td>25</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

The amount of information around us in work and everyday life is constantly growing. As companies grow bigger and their products and projects grow larger, the need of efficient information handling solutions is becoming more and more important. There is a need of information that is easy to retrieve, which is traceable and simple to edit and store as well as fully adaptable to the surrounding circumstances. There is also a need of conformity within all kinds of cooperation. Development of software applications to support workers and consultants in their work is a way to handle these issues, but not all design solutions ensure efficiency and effectiveness. Adapting a user-centered approach in the design process is one step further for producing applications that really add values to its users.

1.1 Background

1.1.1 The Company

Sigma information design AB (SID) is a consulting firm that offers solutions in three specific areas; Customer Product Information, Embedded Design and Information Management. SID’s main customers are Volvo and Ericsson, and for Ericsson GSN-project in Göteborg, SID is providing general solutions for the producing of Costumer Product Information (CPI).

1.1.2 The Problem

Until 2001 the program MS Excel was used to handle the CPI documentation, but as the project was growing the amount of documentation grew bigger, and the need for a system that could handle the large quantity of information was increasing. SID developed a system called Docomotive for CPI-handling, and what used to be the handling of up to 800 MS Excel documents could now more easily be handled in one system with the abilities for searching, editing and storing all kinds of meta-data. Docomotive is a secure web application and has been in use since 2001, and since new functions is needed whenever there is a new project coming up, the number of functions and additional tools is increasing over time. Several programmers have been involved in the development of Docomotive, which can easily be seen in the appearance of the Graphical User Interface. Even though the program has facilitated the work of CPI documentation a lot, the lack of consistence and usability aspects in the program is obvious. Since different parts of Docomotive have been developed separately, they differ in appearance and aspects of usability. Hence Docomotive as a whole is needed to be seen over and designed in a more consistent way, emphasizing usability aspects.

1.1.3 The Task

The purpose of this master thesis is to analyze and improve the interface in Docomotive thus making the application more usable. A second purpose is to produce a guideline for further development of the program. The goal with the master thesis is to:

- Make a usability analysis of all the applications in Docomotive
- Write a report consisting concrete improvement proposals
- Together with Sigma ID choose the most relevant proposals (those who mostly will improve usability and make the system more homogeneous) and implement as much as possible in HTML and ASP within a period of five weeks, to be tested functionally and introduced in Docomotive.
- Make a guideline that describes principles of how Docomotive is to be further designed (this goal was added later and not established from the beginning).

Partners to considerate in this master thesis are:

- Sigma Information Design AB
- Ericsson’s directives for CPI production
- GSN CPI

1.1.4 System description

Docomotive is an internal application that is adapted to aid Costumer Product Information (CPI) documentation, and is used by consultants at Sigma Information Design working in the GSN-project at Ericsson. Docomotive
handles document meta-data, not actual documents. Metadata is structured information that is created specifically to describe another resource, in this case information concerning documents. The actual documents or the sharp revisions ready to be delivered to the customer, are written as SGML files and are stored in Ericsson’s CPI-database, GASK. The relation between the storage of the document information and the actual documents are shown in Figure 1.

![Figure 1. Docomotive’s relation to Ericsson CPI at GSN](image)

In Docomotive a document’s metadata consists of information such as: when the document last were updated, what kind of changes are made in the document, name of responsible technical writer, number of the current revision of the document, status of the document (removed, temporarily removed, not included yet etc.) engaged/unengaged document, clone information, what product and builds the document are in, review information and much more. Even though Docomotive is handling metadata described, the actual (or sharp) documents are linked to GASK and are accessible from Docomotive.

By sending a formatted SGML-file, called SDI, to Ericsson’s system for CPI-generating (DWAXE) with a description of the structure hierarchy, which documents and which revision that is to be delivered, DWAXE generates an ALEX-library (Active Library Explorer). This Zip-compressed file consists of the necessary files in the required file structure, and are now in a PDF format, ready for deliverance to the customers. A system overview is presented in Figure 2.

![Figure 2. System overview and relations to Ericsson’s CPI in the GSN project](image)
The most used tasks in Docomotive are searching documents, edit and store document related metadata and information changes. There are several tools to aid these tasks, such as preDoc, DoG, Templet, GLAPP and CCF-Tool. These are tools for building structures and hierarchies, glossary with allowed and forbidden terms as well as for producing meta-data in a common and consistent way.

2 LITERATURE

2.1 Human Computer Interaction

When developing a new system or application with a graphical user interface, companies often let a programmer without knowledge in usability do the job. The result is usually a functional program, if you just know how to use it. If the system or application is not developed with a usability perspective at all, the users have to spend a lot of time trying to find help in manuals (if there are any), or asking co-workers how to solve the task. If there are other ways to accomplish the task, there is a risk the users won’t use the application or have no use of the specially developed functions in the application at all. The term of usability has its origin in the discipline of human computer interaction (HCI) and the big gain of introducing usability in the design process is that the users of an application can use it without help, thus minimize support and training, and to accomplish the task more effectively.

2.1.1 Diversity of Human Computer Interaction (HCI)

HCI is a multi-disciplinary subject. The design of an ideal interactive system requires expertise in a range of various disciplines; psychology and cognitive science for knowledge of the user’s perceptual, cognitive and problem solving skills; ergonomics for knowledge of user’s physical capabilities; sociology to help understand the wider context of the interaction; graphic design to produce an effective interface presentation; computer science and engineering to be able to build the necessary technology, and so on. Economics and business is also desirable skills to be able to produce a cost effective development plan and for marketing and branding the system or product. (Chen, 2005)

There are no unified theories of HCI, different disciplines concentrates more or less on various aspects of HCI. But they all have one common perspective, that is, to put the user in focus and to design after user needs and how they interact with products and applications in real life, both in work and for leisure. HCI involves the design, implementation and evaluation of interactive systems in the context of user’s task and work. Therefore a mix of expertise in various areas and skills are needed to fit a certain products characteristics and thereby, in the end, find a good as possible solution in the design of Human Computer products.

2.1.2 Goals with Usability in User-Interface Design

A product or application may have a complete set of functionality to fit worker or customer tasks. However, the actual workflow, time spending and cost may not have an optimal solution. One clever design may fit perfectly for some users, but others not so well. Therefore it is important to define user categories and needs as well as include users in the design process. This could either be done by including users in the design process (Luck, 2003) or performing usability tests on prototypes and existing applications. Data gathered from either of these are most valuable for the designer and developers and in turn for managers and purchaser. Ben Shneiderman (1998) states five categories for measurements in human factors that are central for evaluation purposes and these are:

1. **Time to learn** – How long does it take for typical members of the user community to learn how to use the commands relevant to a set of tasks?

2. **Speed of performance** – How long does it take to carry out the benchmark tasks?

3. **Rate of errors by users** – How many and what kind of kinds of errors do people make in carrying out the benchmark tasks? Although time to make and correct errors might be incorporated into speed of performance, error handling is such a critical component of system usage that deserves extensive study.

4. **Retention over time** – How well do users maintain their knowledge after an hour, a day, or a week? Retention may be linked closely to time to learn, and frequency of use plays an important role.
5. **Subjective satisfaction** - How much did users like using various aspects of the system? The answer can be ascertained by interview or by written surveys that include satisfaction scales and space for free-form comments.

All of these are important and succeeding in every one of them is an ideal state of accomplishment. However, in real practice designers often are forced to make tradeoffs, either by economic constraints or lack of expertise knowledge. There are also forced tradeoffs between these categories, for example, if the rate of errors is to be kept at an extreme low level then speed of performance may have to be sacrificed. In other applications the subjective satisfaction, short learning times or rapid performance may be of highest priority. Project managers and developers must be aware of the tradeoffs and make clear which goals are primary. Design guidelines are also useful for communicating ideas and goals of the product or application and to secure common terminology, formats and ideas of the design. These guidelines are more important to establish if the developers and the engineers are changing over time or the application itself has the nature of constant adjustment according to customers needs.

Prototyping with user tests and proper evaluation methods together with an explicit design guideline helps to meet the designers and customers needs which in the end should be measured with acceptance tests to verify its goals.

### 2.1.3 Benefits and Potential Outcome with Usability Engineering

Iterative usability testing during the design process has many advantages. First, a few small tests are more valuable than one large test at the end. The sooner problems are detected, the less expensive to fix them. Second, finding and fixing problems at an early stage means less rework. This not only saves money, it also reduces developers and designers frustration. Third, testing one part of the application may teach valuable lessons than can be adapt to other parts, thus making different parts coherent and development easier. Fourth, changes are more likely to be done if detected early in the design process. Cost and time usually leaves problems to be unsolved if detected late or after development. Only the most critical problems are taking cared of. Fifth, acceptance testing to ensure that deliverables are matching user, and thus, costumers needs (URL http://www.usability.gov/basics/index.html#difference).

Usability engineering also saves money. One of the first methods used to calculate benefits with usability engineering was Clare-Marie Karat at IBM who used it to show a 100-fold return on an investment of a particular software product. In that case, spending 60,000$ on usability engineering thorough development resulted in savings of 6,000,000$ in the first year alone. (Karat, 1994) In short, this method calculates the time saved for the user to complete worker tasks and multiplies this with the number of workers doing these tasks and their average hourly salary.

### 2.2 Interaction Design

#### 2.2.1 The Interaction Designers Role in the Design Process

It is important to understand that interaction design is not conducted by a single person. Interaction design is a design process that, besides the interaction designer, involves usability engineers, system and software developers, and graphic designers. Selling, Sjöberg & Norlin (2002) argues that the strength and role of an interaction designer it to understand the design process and to bring out the strong concepts that leads to exiting and successful products. It lies within the interaction designer’s competence to capture and put into practice the cultural and temporal aspects of users’ values and expectations. Since interaction design is a multidisciplinary field, several different tools and methods are used to extract user demands and technical limitations for finding a suitable solution for both the user and developer. Many of these methods can be derived from User Centered Design (UCD) and HCI. Planning and coordination of knowledge and competence becomes important when directing the design process to desirable goals. Löwgren and Stolterman (2004) argue that it is important to accentuate that the design process is not linear. It is not only an iterative process but a fully dynamic process where all parts of the process influence the others continuously.
2.2.2 Methods and Tools

Interaction design has a lot in common with traditional product and industrial design considering methods used, albeit interaction design focuses on products within information technology. Examples of methods are Heuristic Evaluation (Nielsen, 1994), Cognitive Walkthrough (Clayton, 1994) and contextual inquiry (Hugh, 1998) for finding problems with usability. Another method is Card Sorting that can be used to make system hierarchy visible as well as search for navigation patterns between different parts of an application. In short, this method could be done by placing small size screenshots on a table or on the wall to get an overview of the application hierarchy. Then connections are made between these for visualizing navigation connections.

Interviews and surveys is another important and good way of gathering data from users. However, to have an objective point of view, collecting reliable data as well as asking the right questions is far easier said then done. First, the users may not tell or even know what they want or what aspects in the design that causes usability problems. Second, it is easy to make the mistake in forming questions that leads to expected answers and thus not giving any additional information. Third, the chosen information persons may not be representative for the population of users using the application. And fourth, surveys and questionnaires may provide quantitative data by including a greater number of information persons at a lower cost. Interviews on the other hand, provide qualitative data from fewer interview persons since it takes considerably longer time to make appointments and performing the interview as well as to analyze collected data. (Karlsson, 2005)

One way of finding out how users interact with a product in their natural context is to perform observation studies. In this case, when the subject of investigation is application software, observation with Think-Aloud Technique (Lauesen, 2005) is a formidable way of gathering data of how the users do in action. Information such as number of clicks to perform a task, how users perceive and scan information provided on screen and which ways they navigate to achieve their goals can be observed this way. Valuable information about how users think when performing tasks, solving problems and interpreting information can be gathered with the Think Aloud Technique.

When all necessary data is gathered and analyzed there are several methods of converting the data into design concepts and proposals. Paper prototyping and Rapid Application Prototyping (RAP) is commonly used to present design proposals. When using paper prototyping quick sketches are made for communicating layout and interaction sequences. Rapid Application Prototyping is much the same, but this time the prototype consists of, for example, pages in HTML that are linked to each other. In this way it is possible to simulate specific action sequences with a basic design and layout proposal.

2.2.3 User-Centered Design (UCD)

Within HCI and UCD many have tried to develop standards or principles on how to incorporate users and usability aspects in the development process. Bevan (2001) describes four categories that should be considered for any company that are incorporating usability aspects. First, the use of a product should be evaluated - functionality, efficiency and satisfaction for a user in context when using the product. Second, evaluate the user - interface and interaction with the product. Third, evaluate the design process in which a product is developed. And fourth, examine the possibilities of a company to adopt a user centered process.

Another way is to hire interaction designers and/or usability experts for evaluation of one or several of these categories. After evaluation decisions could be made how to incorporate usability aspects in the design process, for example, creating design guidelines or propose a modified design process.

2.3 Cognitive Perception

“Cognitive psychology (as a science) is unified by a common approach based on an analogy between the mind and the computer. This information-processing approach views the mind as a general-purpose, symbol-processing system of limited capacity.” (Eysenck and Keane, 2000) The gestalt approach based perceptual organization on a group of laws (Eysenck and Keane, 2000). These laws can be used when designing user interface that consists of large amount of objects and two of them are illustrated below in figure 3 and 4.
Figure 3. The law of proximity – Objects that are close to each other appears to be related. To the left the circles form vertical lines and to the right, horizontal lines.

Figure 4. The law of similarity – Objects that are similar are seemed to be grouped, even if they are at the same distance from each other. The circles and squares form vertical lines.

This can be very useful when structuring information in long lists. By grouping objects with the help of space, shape and color, the percept of complex information can be structured to aid human perception.

2.3.1 Color

The using of colors in a graphical user interface can make an interface more attractive, more intuitive and easier to understand. When used in a careful and conservatively way, colors can improve task performance and make the understanding of a system easier (Shneiderman, 1998). But danger in misusing colors in a user interface is high; when used in an incorrect way colors can for example draw the attention to the wrong thing on a screen or make a confused impression on information presentation on display. Shneiderman proposes some rules for the design and use of colors in an interface in a proper way:

- Use colors conservatively
- Limit the number of colors
- Recognize the power of color as a coding technique
- Ensure that color coding supports the task
- Have color coding appear with minimal user effort
- Place color coding under user control
- Design for monochrome first
- Consider the needs of color-deficient users
- Use colors to help in formatting
- Be consistent in color coding
- Be alert to common expectations about color codes
- Be alert to problems with color pairings
- Use color changes to indicate status changes
- Use color in graphic displays for greater information density
There is a number of Western (Western Europe, US and Australia) color connotations as identified by Aaron Markus in his book *Graphic Design for Electronic Documents and User Interfaces* (Markus, 1992). These should be considered as guidelines and not as strictly as is. Colors connotations can vary across different cultures and even have the opposite meaning and they may not even fit in some situations, therefore they should be considered as just guidelines.

- Red – Danger, hot, fire
- Yellow – Caution, slow, test
- Green – Go, okay, clear, vegetation, safety
- Blue – Cold, water, calm, sky
- Warm colors – Action, response required, proximity
- Cool Colors – Status, background information, distance
- Grayscale, white and blue - neutrality

The using of color coding can be very effective when appropriate used, however caution have to be made since inappropriate use easily can cause the opposite effect, and leaving the user confused and disoriented.

### 2.3.2 Icons

Barr, Noble & Biddle (2003) identifies three types of signs that are particularly interesting when designing icons: iconic signs, indexical signs and symbolic signs. “They are all related in that each type is individuated by the representamen’s relation to the object. In terms of computer icons, this means that the sign types are divided according to how the icon relates to its underlying functionality.” An iconic sign occurs when the sign relates to the object through the process of resemblance. For example, the icon for a document on the desktop is an image of a piece of paper which specifically resembles the object in question. An indexical sign occurs when the sign relates to the object through causation. The “Print” icon in Microsoft Word is an indexical sign because the image of a printer represents the cause of the desired action of printing. Finally, in a symbolic sign the representation relates to the object purely through convention. The image for “reload”, along with its textual capitation, in the Mozilla browser is an example of this (Figure 5).

![Figure 5. The document, print and reload icons as iconic, indexical and symbolic signs](image)

Marcus (1992) applies semiotics as a guide to four levels of iconic design:

- **Lexical qualities.** Machine-generated marks such as pixel shape, color, brightness, blinking.
- **Syntactic.** Appearance and movement – lines, patterns, modular parts, size shape
- **Semantics.** Objects represented – concrete versus abstract, part versus whole.
- **Pragmatics.** Overall legible, utility, identifiable, memorable, pleasing.

When creating an icon he recommends to start with quick sketches, pushing for consistent style, designing a layout grid, simplifying appearance, and evaluating the design by testing with users.

### 3 METHOD

The human centered design processes for interactive systems, ISO 13407, provides guidance on achieving quality in use by incorporating user centered design activities throughout the life cycle of interactive computer-based systems. The process describes user centered design as a multi-disciplinary activity, that incorporates human factors and techniques with the intention of enhancing effectiveness and productivity, improving human working conditions, and counteracting the possible adverse effects of use on human health, safety and performance (URL [http://www.usabilitynet.org/tools/13407stds.htm](http://www.usabilitynet.org/tools/13407stds.htm)).
The human centered design processes for interactive systems includes four user centered design activities, and these are to:

- Understand and specify the context of use
- Specify the user and organizational requirements
- Produce design solutions
- Evaluate designs against requirements.

The iterative nature of the activities is described in Figure 6.

![Figure 6. User centered design activities throughout the life cycle of interactive computer-based systems.](image)

In this project, the ISO 13407 process has been applied. Several small iterations can be made mainly within step 4, and also between step 4 and 5. When larger changes have to be made, it may be necessary to go all the way from step 2 to step 5.

### 3.1 Planning the Process

Before the actual work began, the need for a plan and a schedule was necessary. The schedule was developed in cooperation with the supervisor at Sigma. Since it was very hard to estimate the time consumption for the different phases in advance, the number of weeks estimated for different phases are approximate. The phases also overlap each other since all steps are conducted in a strictly sequential order. (App. D)

**Introduction, 1 week**

The work starts with an introduction to Sigma and to the process Sigma uses in the projects, called RUP (rational unified process). RUP is a process for development of software, and describes who is doing what, when and how. The introduction phase includes activities like learning more about Sigma by reading company literature about routines, and also to learn more about Ericsson’s routines for CPI-production in the GSN-project.

**Feasibility study, 2-3 weeks**

In the feasibility study the goal is to identify the Docomotive users, why, when and for what purpose Docomotive is being used. Then to get introduced to and learn Docomotive, find out the goal and purpose with the system and to get an overview of how information is handled and stored. The feasibility study includes reading available Docomotive literature.
Choose and design of methods, 1-2 weeks
The next step is to choose and design methods for the expert evaluation and the usability study. Then to determine available users for user tests and user contact. The literature to be studied in this phase is literature in HCI and methods.

Usability study, 6 weeks
The next phase in the project is the usability study. The phase begins with planning and booking the users for the test. The phase includes the implementation of the user test and the expert evaluation. The gathering of data and analyzing data from the tests is to be done, and also to compile the users and the experts’ requests.

Producing design and concept proposals, 4 weeks
In the following phase it’s time for producing design and concept proposals. All the changes in the interface are to be considered and prototyped. There should also be time for one or more iteration of the design proposals. The phase should result in a final design proposal to be implemented.

Implementation, 5 weeks
The next phase is the implementation phase. Now the coding of the changes in the interface is to be done. New icons are to be designed, and color scales chosen. Testing and verifying the solutions are also to be done.

Evaluation, 3 weeks
Implementation of the user test for measuring settled usability criteria, and to analyze, compile and document the result.

The final step is a presentation of the results for involved persons at Sigma Information Design.

3.2 Identifying the Users
The number of users that currently are registered for using Docomotive is 23, but in the near future the number of Docomotive users will grow. The majority of the Docomotive users are situated at Sigma in Göteborg, but there are also some Docomotive users at Sigma offices in Stockholm. There are two different kinds of Docomotive users, technical writers that write CPI-documents for GSN-project at Ericsson, and “builders”, the persons that put together the SDI’s for delivery having the overall responsibility for different builds in the project. At Sigma in Göteborg there are two persons working as builders, and the rest of the users are technical writers. Depending on the users’ role in the project, they have access to different functions in Docomotive. Users only have access to the functions needed in their work. Technical writers have access to functions such as the search functions and editing metadata information and the builders have extensive access to the program. The Docomotive users differ a lot in their experience of working with the program, some users have been working in the program since it first was introduced in the company, and some users are beginners. The aim in this master thesis was to test both experienced users, and users that hadn’t been working that much with the system. Even though most of Docomotive users are situated at Sigma’s offices, it is possible for the users to work in Docomotive from home or at customers’ offices, since the program is a web application.

3.3 Choosing Methods for the Usability Analysis
In usability testing there are both empirical and theoretical methods used in the analysis of a product, in this case a software application. Common ways of evaluating user interfaces are empirical methods of which user testing is often used. A problem with testing real users is that it’s expensive to recruit users in sufficient numbers. Furthermore, companies’ budgets sometimes impose restrictions that make theoretical methods like inspection desirable as a “discount usability engineering” solution since they are highly cost-effective. Research have shown that usability inspection methods are able to find usability problems that are overlooked by user testing, but user testing also finds problems that are overlooked by inspection methods (Nielsen, 1992). This should mean that the overall amount gathered usability problems can be found by combining different kind of methods.

Examples of inspection methods, which don’t include user participation, are methods such as Heuristic Evaluation (Nielsen, 1994) and Cognitive Walkthroughs (Clayton, 1994). Since a Cognitive Walkthrough means going through the whole GUI, a Heuristic Evaluation is a more appropriate method for finding problems in the GUI of a large application.

There are a lot of usability methods that can be applied in a product developing process, and it is important to choose methods that will serve your purpose in the specific parts of the project. When examining the users’ general opinions of a product, and to get a general understanding of the users’ attitude towards a product, a
survey is a good method to use. When using a survey, quantitative data collected from a large number of users can be gathered and statistical analyzed.

Many aspects of usability can best be studied by querying the users, especially concerning issues related to the subjective satisfaction of the users and their possible anxieties. But since surveys, questionnaires and interviews are indirect methods, this technique does not study the actual user interface, it only collects the user opinions of an interface (Holzinger, 2005).

Advantages of indirect methods include that subjective user preferences and satisfaction can easily be identified. A disadvantage with indirect methods is that they identify fewer problems than other methods (Holzinger, 2005). Usability inspection needs to be combined with usability test methods and thus indirect usability tests should be combined with direct usability tests. One example of a direct usability test is observation with or without the Think Aloud Technique. The direct usability tests are needed to understand the user’s task and the users’ abilities and capabilities. One of the most powerful ways of getting user information is through observation of users in their work context, and developers who already thought they had a good idea of the users’ work and needs are often amazed at how much they learn through observation. Information gathered from methods like observations, are information that cannot be discovered any other way (Susan M. Dray, 1998).

The methods chosen to be conducted are a survey, interviews, observation with Think Aloud Technique and Heuristic evaluation. Although some data are expected to be gathered from all of the methods, the methods are chosen to complement each other, and thus covering as much as possible of the potential problem space (Figure 7).

Figure 7. Several methods used to cover as much as possible of the potential problem space. The white areas in the circles represent problems found within that method, but not in the others. The blue areas covering two or three circles represent same problems found in two or three methods, respectively. Same usability problems found in two circles or more (blue area), strengthen the fact that it is a usability problem.

3.3.1 Feasibility study; Context of use, Interviews and Literature study

In a feasibility study the purpose is to collect detailed information for essential input to user requirements and the planning of further work. In this project this information is obtained from the users, from the developers of the product and by literature studies. Questions to be answered are:

- Who are the intended users and what is their task in the project? (Why will they use the system? What is their experience and expertise?)
- What are the technical and environmental constraints? (What types of hardware will be used in what organizational, technical and physical environments?)

(URL http://www.usabilitynet.org/tools/context.htm)
To find out more about the Docomotive users, the work started out with a context analysis with the aim to find out who the users are, what experience they have, their role in the projects, why they use Docomotive and what they use it for. For that reason two interviews were made and one technical writer and one builder were interviewed. Persons that were chosen for the interviews were experienced and competent, and could give much information about specific tasks in Docomotive. Since the opinion of a product or a software program is subjective, it was necessary to interview persons performing different tasks in Docomotive. The visual angle of the program mostly seems to depend on the user’s role in the project, and no single user had knowledge in all the applications and functions in the program. By interviewing these different “key users”, the picture of the program became clearer, although a consciousness of the fact that Docomotive may be a part of a complex system arose.

It was also necessary to find out more about the program, where information is stored and how current and related information is handled in databases and by whom. The aim was also to figure out possible technical limitations, and limitations within the organization. To get an overall picture of how Docomotive worked and how different types of information were handled in different databases, two persons with knowledge in the system administration were interviewed.

Since there isn’t any complete and understandable overview of the system as a whole available, the next step was to create a system overview (Figure 1 chapter 1.1) with all database relations. To be able to develop a system overview, the next step was a literature study in Docomotive material, and material about the company and working process.

Now there was time to test and learn Docomotive with all the functions and applications. A problem with the testing of the program is that it is not possible to undo changes in the program. The consequence when working with some of the applications is that e-mails automatically are being sent to involved persons in that project. To avoid interfering with the daily CPI-documentation work at Sigma and Ericsson, the test of Docomotive had to be made in a cloned version of both Docomotive and the relevant databases. This made it possible for the investigators to freely use every function and application in Docomotive, avoiding interfering in the work of CPI-production at Sigma and Ericsson.

3.3.2 Survey

As a first user test, a survey was chosen. Surveys are a good alternative to collect information from a large number of users. Unlike for example an interview, a survey is an indirect question method where an investigator via a form with questions gathers information from users. A well designed survey can give a picture of who the users are, and what they think of a specific product or software program, their priorities of existing problems etc (Karlsson, 2005).

A survey is designed with closed or open questions. Closed questions means that there are fixed answer alternatives, and open questions mean that the users are allowed to answer in an open way, spontaneously giving their comments. A survey could be designed with only closed or only open questions, but a mix between those two alternatives is preferable (Karlsson, 2005). Although a survey with only closed questions makes it easier for the user to fill, that kind of survey tends to give a too narrow picture of the users. A survey including only open questions give the user more freedom to give their opinion, but since it is more work to fill it can result in fewer answers. By designing the answer alternatives as a scale, the survey can be used to get a measurement in the users’ attitude or opinion of a product.

There were two primary purposes with a survey as a first user test. One purpose was to find more information to be used as a basis for designing a further user test. Another purpose was to collect information about attitudes towards functions and the program as a whole, and what kind of applications and functions the users find usable, and what applications and functions the users find unnecessary and not very usable.

The formulating of the questions in surveys is very critical. A formulation that appears very clear to the person designing the survey is often misunderstood by the person answering the question. It’s important to avoid ambiguity, leading questions and long complicated formulations in the designing of the survey. One approach in trying to avoid these kinds of pitfalls is to test the survey on some persons before it is distributed to the actual target group (Karlsson, 2005).

Other things to consider when designing the survey are that the layout is important. An insufficient layout means that the survey is considered complicated and this can lead to fewer answers. Since the using of a survey is a
method where the contact with the users answering is very small, it means that the answering frequency often is low. One way to increase the answer frequency is to follow up with reminders (Karlsson, 2005).

Some strength and weaknesses with surveys are (Mehlenbacher, 1993):

Strengths:
- Inexpensive to administer
- Can be administered quickly
- Maintain respondent anonymity
- Generates quantitative data

Weaknesses:
- Relies on reconstructive data, i.e. on post user experiential accounts of problems with the system
- Encourages response bias, i.e. situations where it is difficult to tell whether respondents to a given survey are representative of the population at large

3.3.3 Heuristic Evaluation

Heuristic evaluation (HE) is a usability inspection method for identifying usability problems and improving the usability of an interface design by checking it against established standards, or “heuristics” (Holzinger, 2005). A great advantage with this method is that HE is a very cost-efficient method, confirming its value in situations where limited time or budgetary resources are available (Nielsen, 1994).

Even though HE seems to be the most used method of the usability inspection methods, and often when using HE a lot of problems are found. Unfortunately, about half of the problems are false in the sense that they don’t cause problems to real users. Which ones this false problems are is hard to know, and furthermore HE in this cases misses about half of the severe problems that real users encounters (Lauesen, 2005). Lauesen in User Interface Design: A Software Engineering Perspective (2005) jokingly refers to this as the first law of usability; HE has only 50% hit-rate. This “law” is quite controversial and many HCI specialists consider the HE to be the best evaluation technique.

The goal of heuristic evaluation is to find usability problems in an existing design (Nielsen, 1992). Originally, HE was developed as a usability engineering method for evaluators who had some knowledge and experience in usability, but were not necessarily usability experts. Later it has been shown that usability experts are better at finding usability problems by HE, than those without expertise in usability principles (Nielsen, 1992). Even though it is a great advantage if the evaluators are experienced in usability, non-experts can be appropriate at times, depending who is available to participate in the method (Holzinger, 2005).

The typical approach of the Heuristic Evaluation is for each of the individual evaluator to inspect and go through the interface alone. The evaluators use their subjective judgment and earlier experience (Lauesen, 2005). After the evaluation is completed the evaluators communicate and discuss their findings. The evaluators come up with a list where each problem only is mentioned once, and also, problems mentioned by only one evaluator must be on the list. This gives the evaluators a heavy job reviewing each others list (Lauesen, 2005). It’s important that the evaluators inspect the interface alone to ensure independent and unbiased evaluations (Holzinger, 2005).

Recognized usability principles that can be used as heuristics are for example Nielsen’s usability heuristics (Nielsen, 1992).

3.3.4 Observation (Think-Aloud Technique) with complementary interviews

Even though a lot of problems can be found in a Heuristic evaluation, the most effective technique to find usability problems is to do a usability test (Lauesen, 2005).

There are several methods for usability testing, and the most common methods are thinking aloud, field observation, and questionnaires (Holzinger). Thinking aloud may be the single most valuable usability test method. The method involves having a user thinking loud while using the system. By verbalizing his or her thoughts, the test user enables the investigators to understand how he or she views the system, which makes it easier to identify misconceptions and problems in the interface. Advantages with the method include revealing why users act in a certain way in an authentic context, and can thereby provide valuable information. The method is useful early in the process, to collect qualitative data for user requirements specification.
Observation means that the examiner on his or her own observes those events that he or she is interested in. The observation is registered by either a written protocol or a video camera. The method can be used in a real environment, for example on a workplace, or in an arranged situation in a laboratory environment. Observation methods are performed to study the user’s actual behavior and possible problems when using a product.

Observations in an authentic environment can be made to get a general picture of the user, the product and the using of the product. Therefore, it can be done in the beginning of a project, as much as to be used as an evaluation of a product later in the project.

Observation studies often result in valuable and detailed information of the user’s behaviors and management of a product or a software program, but since there is a risk in over interpret collected data; it can be valuable to combine the observation with a complementary interview. Combined with an interview, where questions about why a person makes a certain choice and how he or she experiences the situation are asked, a more correct and exhaustive picture can be made (Karlsson, 2005). Observation may be either direct, where the investigator is present during the task, or indirect, where it is viewed by some other means such as a video recorder. The approach of a direct observation allows the investigator to focus on specific areas of interest, and the indirect observation captures activity that would otherwise have gone unnoticed.

The time it takes to perform an observation, is dependent on the product and the situation that is studied. If it is a single work moment that is to be examined, some minutes of video filming will be enough. When observing behaviors when users are working in a program as a whole, the video filming will of course be much longer. In that case it is important to select particular parts of situations that are to be observed. The analysis of the video film will take at least twice as long as the recording, and often it takes 3-4 times as long (Karlsson, 2005). When planning a usability test it is important to find test users that represent the typical users.

### 3.4 Conducting the Usability Methods

#### 3.4.1 Feasibility Study

The context analysis in the feasibility study started out with two interviews. Persons that were interviewed were one technical writer and one builder. The interviews had an open unstructured approach, and a tape recorder was used. The interview with the technical writer went on for one hour and twenty minutes, and the interview with the builder went on for one hour and fifty minutes.

The interviews took place at the participants own work stations, and a computer with Docomotive was being used as a mediating object. Two interviewers were present and lead the questions in to relevant issues. The participants were asked questions about their role and specific tasks in the GSN-project, and asked to show what specific functions they are using, and how they are using them.

There were also several informal interviews made with a system administrator that also is a project leader, and therefore has the knowledge in both organizational and system administrational matters. Since these interviews were sporadic and numerous, they were not being recorded. The purpose with these interviews was to get a picture of the system as a whole. To get an overview of the hierarchy of Docomotive, a card sorting method was conducted. After printing screen dumps of all pages and windows possible, they were sorted out in a tree structure. In that way all possible ways of navigating could be seen, and that made a good overview of the different parts of the program.

#### 3.4.2 Survey

The survey was designed with a mix between open and closed questions. All the closed questions had five answering alternatives, in a scale graded from one to five. (App. C)

The first four questions are closed alternative questions concerning the users’ opinion in degree of usability of the search functions. The answer alternatives were graded from 1 – 5, where 1 was the alternative for “I don’t agree at all” and 5 the answer alternative for “I totally agree”. Question number five was a complementary open question where the user where asked to state if he or her missed something, or wanted to change something concerning the search functions.

Question number 6, 8 and 10 concerned the frequency of some function being used, and the closed answer alternatives were graded from 1-5 where 1 was the alternative for “never”, 2 for “< 1 times per week”, 3 for “1-
5 times a week”, 4 for “1-5 times per day” and 5 for “> 5 times per day”. All of the questions were followed up with an open question concerning for what main purpose the function was being used.

Questions number 12, 14, 16 and 18 concerned the users opinion in degree of usability in some tools in Docomotive, and both the closed answer alternatives and complementary questions was designed just like the first four questions.

In question 20-27, the users were asked to estimate the usability in functions concerning user adapted information. The closed answer alternatives were graded from 1-5 where 1 was the alternative for “not useful”, and 5 the alternative for “very useful”.

Question 29-34 concerned the program as a whole, like the using of colors and icons, and the last question, 35, was an open question where the users were asked to write down if there is anything they think is missing, would like to change or improve in the program.

The survey was designed so that it shouldn’t take more than 10 minutes to fill in. The survey was made in the program Adobe Acrobat reader, and if the user had the program installed, they could fill in the survey on the screen, save it and then send it back in an e-mail. Otherwise the users had to print it and return it in a paper form. The survey was mailed out to all Docomotive users, both in Göteborg and Stockholm. The users got a week to answer the survey, and then a mail was sent as a reminder.

3.4.3 Heuristic Evaluation

The heuristic evaluation was conducted using Nielsen’s heuristics (Nielsen, 1992). Two usability experts were individually inspecting and going through the user interface. After the evaluation was completed the evaluators discussed their findings and wrote them down on a common list. It took around one day and a half to perform the HE and about the same time to compile the list of problems.

3.4.4 Observation / think aloud method with complementary interviews

Five users were observed in the usability test. The users participating in the observation were three technical writers and two builders. The tasks given and the questions that were asked to the technical writers and the builders were the same, but some tasks were added for the builders.

Two interviewers were present, whereof one acting as a moderator conducting the test, and the other taking care of the technical parts like filming the screen. A computer was set up in an empty room with no distraction elements. It was also important for the participant not to have any co-workers around. To prevent distraction and to prevent the participant to feel filmed, a second screen was connected to the computer. This screen showed the same view, and was placed so that it could be filmed during the observation. The filming also included recording the voice of the participant, for further analysis.

The method was conducted as followed; first the participant was given a short introduction of the method’s purpose, how long the test was going to take, and that the method wasn’t to be considered as a test to evaluate the user, but the actual program. The participant was told to think aloud when solving the tasks given, and to explain what he or she specifically was doing in the program. The user was also informed that he or she was anonymous, and that no one but the interviewers was going to take part of the information concerning the specific user. To warm up and to practice a little bit in thinking aloud, the participant first got a test task. The user then was given the actual tasks one at a time, and when performing the tasks he or she explained how it was performed. The moderator sometimes asked the participant to explain further why a task was performed in a certain way.

After performing the observation, the participant was asked about his or her general opinion of the program, if he or she had any complementary information to add.

The observations took 45 minutes to one hour to perform.

The analysis of the films took about three times longer to do than the length of the films. A list with potential problems in the program, both interfacial and functional, was written down. The two investigators analyzed the films individually, and then together compiled a list of problems.
3.4.5 Telephone Interviews

As a complementary to the survey, two telephone interviews were made. Since there was no possibility to perform a usability test of the Docomotive users in Stockholm, telephone interviews were made. Two telephone interviews were made at Sigma offices Stockholm, where Docomotive hasn’t been much in use yet. One of the persons hadn’t used Docomotive almost at all, and the other one had used Docomotive some time ago, but not at the present. The interviews were made with a loudspeaker telephone, where one investigator was interviewing and the other taking notes.

4 RESULT

4.1 Gathered data

4.1.1 Survey

The response frequency of the survey was 70 % since 12 out of 17 Docomotive users filled in the survey.

The answers from some questions stood out from the rest, thus indicating a common opinion and are presented below.

In question 1: “The search function for searching documents is easy to use”, data showed that the users thought the search function was quite easy to use. (Mean: 3.8)

In question 18: "The tool "CCF-Tool" is easy to use”, data showed the users were not very satisfied with the “CCF-Tool’s” ease of use. (Mean: 2.1)

In question 23: "How useful do you find the information field “Last Saved/updated Documents?” data showed the users didn’t find the information field “Last Saved/updated Documents” very useful. (Mean: 2.6)

In question 24: "How useful do you find the information field “Locked Documents and pictures?” data showed the users didn’t find the information field “Locked Documents and pictures” very useful. (Mean: 1.9)

In question 30: "The color coding showing a documents place in a structure (in the “Areas” tab) is distinct and obvious to perceive”, data showed that the users didn’t think the color coding was very distinct and obvious to perceive. (Mean: 2.5)

In question 33: "The icons used in Docomotive are distinct and representative for the activity it stands for”, data showed that the users didn’t think the icons were very distinct and representative for the activity it stands for. (Mean: 2.0)

Commentaries from the survey showed in what purpose different tabs, tools and functions were used, what the users missed in the program, and what functions or tools they would like to see improved. Some of the commentaries were improvement proposals in general, and some were very specific. Examples of commentaries from the survey were:

- “I would like the document numbers in My Central to be clickable links to the stored documents in GASK, like in the search-tab”
- “Maybe all the comments in the comments field in the library card automatically should be marked with date and name of the user writing the commentary”
- “Docomotive should have a more professional appearance”
- “I would like a better looking color scheme, with less pastel colors”
- “I would like to change icon for ”Engage this doc” since I don’t see what it represents”
- “I would like more homogenous icons in the same style and size”
4.1.2 Heuristic Evaluation

The Heuristic evaluation turned out to be an extensive investigation since several problems were found several times in the interface. The HE resulted in answers to questions concerning problems in the interface regarding e.g. navigating, feedback, understanding, using of icons and colors, and consistency in the system. The investigators wrote down the answers to questions like:

1.8 Is there some form of system feedback for every operator action?

When the answer was “no”, the evaluators specified where in the system that problem was appearing and in what way it was showing. In the question as the one above, the investigators explained and specified the problems;

- There is no feedback when clicking the little sun icon in a documents list, that selects the document to be used in Dog.
- When editing the document sorting in the library card for a document (in Areas) and click “close”, there is no feedback telling if changes are saved.
- Same thing happens when editing which builds a document is included in.
- There is no feedback (that we have seen) when generating a SDI/PRI in Alex.
- There is no feedback when you accidentally lock a document (when closing a library card in the upper right corner).
- When engaging a document, the side doesn’t load automatically so that you see that the engage icon has changed. The only kind of feedback you get when engaging and un-engaging a document that a little white window popping up for a millisecond.

During the Heuristic Evaluation a lot of problems in the interface were found.

Examples of problems concerning consistency are:
- Not every display begin with a title or header that describes screen contents
- There isn’t a consistent icon design scheme and stylistic treatment across the system
- The menu instructions, prompts, and error messages don’t appear in the same place(s) on each menu?

Examples of problems concerning the use of color and icons are:
- Not every single, selected icon is clearly visible when surrounded by unselected icons
- The icons are not concrete and familiar
- Excessive detail in icon design hasn’t always been avoided
- There are more than four to seven colors, and they are near along the visible spectrum

Examples of problems concerning feedback are:
- There is not always visual feedback in menus or dialog boxes about which choice the cursor is on now
- There isn’t always some form of system feedback for every operator action

Examples of other problems are:
- There are not always appropriate response times to the tasks
- There is a multiword field label placed vertically (not stacked horizontally)
- Some prompts, cues, and messages are not placed where the eye is likely to be looking on the screen

The heuristic evaluation found an extensive overload of problems in the interface.

4.1.3 Observations (with Think-Aloud Technique)

The observation of the five users showed how they actually act when solving a task. Typical problems in the interface found were problems concerning understanding and navigating.
Tasks given to the participants were specific tasks for the users to solve. Some examples of tasks given to the participants were:

“Find the document Alarm and Event Filter in SGSN R7, and find out if the document is cloned. If it is cloned, from which document is it cloned and is there other children”? Is the document itself a ”parent”?"

Things observed in the task above were:
- Several users (4 out of 5) were not aware of the clone function, and they couldn’t find it without help from the investigators.
- One of the participants tried to solve the task in another way, and find some, but not all information needed.
- Some comments on the color use in the clone function were made.

”You have just performed a formal review and an internal review on the document you recently worked with, and now you are going to inform this has been made. Show how you will perform the task.

Things observed in the task above were:
- All the users find the window where the reviews are done, and they do know how to use the function.
- One user says that he doesn’t always know what happens within the system, he wants feedback for example when an e-mail is being sent.
- One user comments that she is not sure in what form the date is to be filled in, and wants the formatting indicated in the field.

There were two tasks that concerned the glossary application “GLAPP”:
“Find out what AoC is short for. Find a term that is used in SASN but not in SGSN G”

Some of the things observed in the task above were:
- Three users hadn’t used the glossary application before, they look in a glossary document instead.
- Two users don’t understand what the checkboxes are for, and don’t know which one to tick off.
- One user wants to know who has decided the term is in the list, who listed it.
- All of the users think the list is taking to long to load.

The users also sometimes spontaneously gave proposals to improvements to the system. In the task above (the glossary list “GLAPP”) the proposals concerned things like:
- A search function in the glossary would be good
- It should be faster to load, and give feedback that it is loading
- The icon for GLAPP should be more intuitive
- There should be no vertical text

Other problems that were found in the interface were problems that not actually concerned the tasks that were given to the participants, nor commented. These problems were observed by the investigators, either directly or when watching the films. Example of a problem like that was how the users handled the windows opening (windows like “edit documents” window and “edit CCF’s). Since these windows are not opening on the same location, are not fully expanded vertically or horizontally, and are sometimes hiding relevant information, the users spend a lot of time scrolling both vertically and horizontally and moving the windows around.

4.1.4 Telephone Interviews

One person interviewed hadn’t really worked much in the program due to organizational issues. The user requested better instructions for example in the popup texts. She also would like some kind of help file available in the program. Both persons interviewed requested a more consistent approach and professional touch to the program as a whole. They both would like to see lesser colors in the document sorting in Areas, and a more consistent icon design.

4.2 Analysis

The participants in the user tests seemed very motivated to give their point of view concerning Docomotive. They were very helpful both when filling in the survey and in the observation and gave a lot of commentaries
and improvement proposals. Some problems in the interface were discovered in all methods, while others were discovered within one or two methods.

4.2.1 Navigation

The observation discovered some navigation problems in the interface that users were not aware of. An example of a problem like that was that the users spend a lot of time rearranging the windows opening in the application, and to view the content in the windows they also had to use the scrolling lists in the windows a lot. Data from the heuristic evaluation also pointed out that since the windows are not fixed, there are problems with the navigation concerning windows opening.

The observation also pointed out problems in the navigation and understanding of some functions and tools:

- The clone function was very hard to find, even when the users were aware of its existence.
- The users found the PreDoc function hard to use, it was not very intuitive and the help provided was not very supportive since some of the users didn’t understand what it said.

The HE showed that some windows didn’t have titles, which complicated the navigation. User tests and the HE showed that the overview of the document sorting in Areas is very hard to perceive.

4.2.2 Feedback

Data from the HE and the observation pointed out lack of feedback in the program. The observation showed that the users found it frustrating not always knowing what is going on in the program.

Observation showed there is no feedback, but should be, when:

- An e-mail automatically is being sent out (this happens for example when a change in the document information has been done)
- Choosing a document as a reference in the function PreDoc
- Generating a “SDI/PRI” in the tab Alex.
- Building a new template in function Templet, saving a template in Templet
- Locking a document

The HE showed an example of insufficient feedback. When engaging or un-engaging a document in the list, the page has to be reloaded before that can be seen.

4.2.3 Language use

Result from the HE showed that the use of language in Docomotive is not always consistent. For example the grammar is not consistent in the tabs, and in some titles both whole words and abbreviations are being used. The HE also discovered inconsistent use of the title “Area”; in the columns in the document list, showing which Area a document is in, the title says “tab”. Another column in the document list actually has the title “Area”, but is showing which category the documents are sorted in. There is also inconsistent use of the document sorting in the window for editing the document information.

4.2.4 Icons

Observation showed that the users didn’t have common terms for the icons, and they found some of the icons hard to perceive and relate to. The icon for viewing the clone information (a plus sign) was not very intuitive. The observation showed that the users found the edit documents icon (a hammer and a screwdriver) and the icons for engaging and un-engaging documents (a foot cuff) too small and detailed to actually see what they represent (Figure 8).
Figure 8. The edit document icon and the icons for engage and un-engage a document.

The heuristic evaluation found lack of consistency in the icons appearance, size and level of detailing. Also data from the survey pointed out problems with the icons. Data show clearly that the icons have to be redesigned in the same style, color use and size (Figure 9).

Figure 9. The icons differ in size, style, color using and level of detailing.

4.2.5 Colors

Data from the observation, the survey and the Heuristic Evaluation showed that the use of colors in the tab “Areas” is problematic. Since three different types of color coding are being used, the use of colors is extensive. Especially the colors showing a documents place in the structure is confusing and hard to overview.

Figure 10. Extensive use of colors in the “Areas” tab
The Areas tab (Figure 10), shows the document sorting. The colors black, purple and grey are being used for this purpose. There are also colors representing a document’s status, and the colors blue, green and two shades of pink are being used for this purpose. The color red is being used to show if a document is set to “engaged”. There are two different categories in which a document can be sorted, and the titles of the first category in the figure are “General” and “Description” (Figure 10). The second category of titles is “Gb over IP” and “Counter description”. Documents can also be sorted in different types, and in the figure five documents are sorted in the type “Alarm and event descriptions”.

The different document statuses now are five;

- Temporarily removed document
- Removed document
- Don’t include yet
- Non-Alex document
- Building block

In the future the number of types of document statuses is supposed to be extended, and when defining the new color scale for the document statuses this has to be considered. Therefore more than five different colors for status have to be defined.

5 DESIGN PROPOSALS

Several design proposals were made, most of them in Photoshop, and later evaluated and chosen for implementation. When designing the proposals, several iterations were made. The iterations were performed either by the investigators themselves, with the users or in co-operation with the supervisor at SID. After the iterations, new design proposals were produced. Some of the new proposals were being made in Photoshop, but proposals that were close to a possible solution were directly implemented in the application.

This chapter starts by describing problems that need to be fixed concerning feedback, consistency and navigation. Proposals for new design of Areas, Search, the edit document window and the edit CCF window are presented in chapter 5.1.

Numerous problems in the interface concerned feedback, consistence and navigation issues. Many of these problems are not seen by just looking on different parts of the program, but noticed in interaction. Fixing problems like these would make a great improvement concerning usability issues in the program. Proposals on improvement concerning feedback, consistency and navigation are:

- Same type of windows should open on the same location on the screen, and not hide relevant information.
- Windows should be in a fixed width, with the option to resize and scrolling vertically
- There should be pop-ups describing where it leads and which actions that can be made, when the mouse is over an icon or link
- Pop-ups should be fixed, not floating (as earlier)
- There should be feedback of the system’s status whenever lists are taking a long time to download
- There should be feedback when a mail automatically is sent
- If applicable, no more than three windows should be opened at the same time
- The back button should be preserved, whenever applicable, otherwise error message should inform users of errors status and how to recover.
- The cursor should become a hand everywhere the cursor is over a link or an icon
5.1 Proposed Design Solutions

5.1.1 Areas, proposal 1

![Proposal 1](image.png)

Figure 11. The first design proposal of page “Areas”

1. Instead of the heavy use of colors, document sorting now is arranged by the indentation of the documents titles (in the left column). The placement of the document title shows a document’s place in the hierarchy. When the document titles are left adjusted, the documents belong to the first category and when the document titles are indented they are in the second category. Documents under a bold title of description are sorted as a type (same type as the bold title describes) and when not a type, the documents are sorted directly under the second category. In this proposal there is just one type of color coding. This makes a better overview.

2. The use of colors showing a documents status is the same as before. The new colors are the same, but now the colors are web safe and less heavy.

3. The tab no longer consists of a drop down menu; choices of Areas and Builds are now made on the specific side.

4. The red color showing if a document is set to engaged, is now minimized to only be seen in the right column. The color is now softer and less alarming.

5.1.2 Areas, proposal 2

Proposal 1 for Areas (Figure 11), where indentation of the documents titles is being used for document sorting didn’t really work out as well as hoped. It showed that the spacing used for the indentation was too small to get an overview of the hierarchy. When increasing the space, the documents titles in many cases were extending from one into two lines, which made the lists too long.

Instead a new design with a tree structure was considered (Figure 12). This proposal was not made in Photoshop but directly implemented as HTML code. The previous empty space within indenting have in the second proposal lines connecting blocks. A new column and icon for revision history were added, as well as two arrows (icons) for sorting documents by title or number.
Figure 12. The second design proposal of page “Areas” with tree structure.

5.1.3 Search, Proposal 1

Figure 13. The search tab
Figure 13 shows the design proposal of “Search”.

1. The texts on the two sets of tabs in “search” are now black and the grey color of the tabs is lighter, so that the texts on the tabs are more readable.

2. The icons are now grouped in the right column for a better overview.

3. There is now an icon that shows the current revision. When clicking the icon revision history can be seen. The icon makes it visible that it is clickable.

4. The document sorting is now being grouped in the same column for a better overview.

5. The column “Tab” is now being called “Area” for a consistent language and to facilitate navigation.

6. The column for the clone information is removed from the list, to save space. The clone information is now reachable from the edit documents window.

7. To group the search functions, there is now a thin black frame.

5.1.4 Edit Document Information, Proposal 1

Figure 14. The design proposal for “Edit Document Information”
The proposal for the edit documents window (Figure 14) includes:

1. The edit document window now consists of a white background on a checked pattern for better readability.

2. The window now has a clear title that says “Edit document information” for better navigation.

3. The fields for entering information are now aligned, and information fields and functions in the window are grouped for a better overview.

4. The clone information window is now available from here.

5. “Remember”, where delivery information are being edited and viewed are now called “Delivery info” for a more consistent use of language.

6. CCF history information is now available directly on the window, instead of being hidden under “included in builds”.

7. Buttons instead of links are used for CCF-Tool and edit delivery info (former Remember link) for a better overview.

8. Buttons instead of links for the editing of “document sort” and “included in builds”, for a more consistent use of linking.

9. Save and Cancel buttons are grouped and right aligned, and the Clone button is moved a little to the left, for separation.

5.1.5 Edit CCF, Proposal 1

Figure 15. The design proposal for “Edit CCF Information”
The proposal for the edit CCF window (Figure 15) includes:

1. The window has a title that says “Edit CCF information” for better navigation.

2. The window now consists of the same white background on a checked pattern, as the edit document window, for better readability.

3. The fields are aligned for a better overview.

4. The Save and Cancel buttons are grouped in the same way as in the edit document window.

5. The field for filling in the CR title is removed, now the title of the TR is filled in the field TR.

5.1.6 Icons

New icons are all in 16 x 16 pixels or all in 32 x 32 pixels. For a consistent approach of the icons, similar colors are being used. The icons were created so that they are not very detailed.

Several icon proposals for Engaging and un-engaging a document were made. The foot cuff used as a metaphor in Docomotive didn’t seem very intuitive; it is not like the document are being “chained” to the user engaging it, the user only makes others aware of the fact that he or she currently is working on the document. Since it was hard to find a metaphor for the purpose, only one metaphor (a bird cage) was made as a proposal. Instead icons were made as buttons. Examples of proposed icons for engage / unengaged are shown in Figure 16.

![Figure 16. Icons representing engage and un-engage](image)

First set of icons made is a red button (same red as in the right the column when a document is set to engaged) for engaging the document and a green button for un-engage. Second set of icons represents a “U” for un-engage, on a white button, and an “E” for engage on a red button. Third set of icons are a closed cage for engaging and an open cage for un-engaging.

Figures 17-24 are showing examples of other icon proposals made.

![Figure 17. Icon proposals representing “clone information”](image)

![Figure 18. Icon proposals representing “edit document”](image)

![Figure 19. Icon proposals representing “locked document”](image)

![Figure 20. Icon proposals representing “view documents in that build”](image)
5.1.7 Colors

The color coding used in the tab Area where the colors are showing the document sorting is now replaced by the indentation of the document titles. The way of showing the document status is still the same, although the colors are slightly changed. Since the users are used to the present colors, the change rather concerns the depth and strength of the colors. Now the colors are lightened up, and less strong and heavy. Since new types of document statuses possibly are needed in the future, some colors are added for that purpose.

Color for the “building block”-status is now removed, since it is no longer in use. Colors are described in hexadecimals, and are shown in Figure 25.

6 IMPLEMENTED DESIGN SOLUTIONS

The implemented design solutions were chosen either by consulting the supervisor or evaluated against user requirements. Some of these were also tested and consulted with users before selected.
6.1 Main application

![Image of the main application](image)

Figure 26 The main application

Figure 26 is showing the implemented solution of the search page in the main application.

The tabs dividing the application into sections were replaced with a new background image for better readability and the text color remain black at all times. The color coding in bottom of page is visible throughout main application. The orange color for the “building block” status was removed since it is outdated, and the dark blue color for “Removed” was added. The saturation of the colors showing the document status was leveled down to avoid too heavy color impact in cases when a large block of colored documents posts are showing. Icon “GLAPP” was replaced with a new icon.

The popup information as well as titles and headlines were changed where not clearly phrased. For example, the popup information on the “Edit” icon - “Manage” was changed to “Edit delivery information”. The previous version had also a JavaScript function for displaying popup information that followed the cursor. Since these showed up on different locations on the screen annoying the users, this function was removed and replaced by ordinary popup information with `<alt>` and `<title>` tab.
6.2 Page “Areas”

Figure 27 is showing the solution for the tab “Areas”.

The purpose of the “Areas” tab is to show documents sorted by different areas. Every document can be sorted in two levels of depth with a type definition. The colors representing levels in the hierarchy were removed and replaced with indenting. Blocks having the same indentation are now visualized with vertical lines. An arrow next to column head “Title” and “Document Number” were added, this for sorting documents by title or number. Icons are grouped in the right column as in the search page.
6.3 Page “Search”

Figure 28 The search tab showing the document list

Figure 28 is showing the solution for the document list in the search tab.

The four tabs were adjusted so that each search criteria is ordered in the same top-down structure. The search area is also set to a minimum width so that the downsizing of windows is not affecting the layout and thus making radio buttons and text “float”. The same applies for the column width, whereas “Title” and “Document number” don’t resize below a certain width. The column “Revision & history” also have a max length as well as the column for clone information, this to leave as much space as possible to the column “Document sort”. Columns “Tab”, “Area”, “Function” were merged into single column with column head “Document Sort” to make the application more generic and to utilize unused space. The new formatting will show “Project area > Area > Function > Type”, that is the complete document sorting and type for all documents.

The “floating” pop ups, with the help text following the cursor, are removed and popup information is added through the <alt> and <title> tabs. Since the revision history information only can be viewed for a specific build, the column is not seen for the search criteria “Show which build”. All icons, except the “revision history” icon (that is added), are grouped in the last column to the right with the “engage” and “edit” icons. The other icons appear only when applicable. Icons for “Engage/un-engage” are replaced and the icon for “Locked” is replaced with an icon using a more appropriate metaphor.
6.4 Page “Edit Document Information”

Figure 29 is showing the solution for the window “Edit document information”. The checked background for the window was removed and replaced by a white background, to support a good readability of the text blocks. The window now opens fully expanded with the only scrolling option vertically. All the button now have a unified size and the colors red, green and blue are restricted to the “Save”, “Cancel” and “Clone” buttons. The window now has a title describing the content and actions possible. The colors and buttons in the popup windows opening within the window were replaced by neutral colors to support readability and a unified color use. The radio button for “Building Block” was removed since it is no longer in use. The previous pop ups with the “floating” help text following the cursor are removed, and popup information are added through the <alt> and <title> tabs. The texts next to the form filling fields are left aligned.
6.5 Edit Change Content Form (CCF)

Figure 30 is showing the solution for the window “Edit CCF”. The window now consists of a solid light background color as the other windows, and a title and description. The window opens fully expanded with structured information fields and buttons “save” and “close” aligned to the right.

6.6 Icons
The icons chosen are the following:

- Edit document, Edit CCF and Edit build
- Engage a document
- View clone information
- View documents for a build
- View revision history
- View old reviews (yellow)
- Locked document
- Un-engage a document
- View CCF’s for a build
- Open GLAPP
- Start new review (white)
6.7 Colors
Colors chosen are the same as in the proposal, except for the red color showing when a document is set to engage. Even if the red color is meant to alert the users, the color in the proposal is too strong and is flickering.

7 EVALUATION OF IMPLEMENTED DESIGN SOLUTIONS

7.1 Method
An evaluation was made on three Docomotive users, two technical writers and one builder. The users were all asked the same questions while they could look at the new interface design on a computer screen with Docomotive. One user was interviewed at a time, and one investigator was present.
The investigator first showed the new icons printed out on paper. The investigator asked the user to tell if he or she could tell what actions the icons represented, and what he or she thought about the new icons in general.
The investigator then asked what the user thought about the new design of “Areas”, “Search” and the windows “edit document” and “Edit CCF”, and what he or she thought was better, worse or no different concerning the new design.

7.2 Result
7.2.1 Areas
Two persons liked the new tree structure in Areas a lot. They thought the tree structure made a better overview and that it now was easier to perceive the document structure.
The third person, the builder, thought that the new tree structure was not very good since there are lines interfering with the overview.
All the users liked the fact that the colors used for the document structure now are removed.

7.2.2 Search
Two persons thought that the column with the tools should be in a more centered position, instead of being in the right column. Two persons thought that the column for "clone information" should be called "clone history" instead. All persons asked liked the new column for document sorting, and found it easy to overview.
All persons asked liked the fact that all icons now are grouped in the right column.

7.2.3 Edit documents / Edit CCF
All interview persons liked the new edit buttons better than the link used before, and they liked also the fact that all windows have titles. The users thought that the new grouping of information was good, and they also liked that there is a solid background color although two persons thought it should be another color than white.

7.2.4 Icons
Two users could figure out the actions of the following icons;

Even though they liked that there was an icon for viewing revision history, the users thought the icon was an icon for "help". Two interview person thought the new icon for opening the glossary application "Glapp", was an icon for sorting documents.
8 THE DESIGN GUIDELINE FOR DOCOMOTIVE®

This guideline is developed as deliverance to Sigma Information Design. The purpose of the guideline is to ensure consistency and usability in the web application Docomotive, and to assist developers and programmers in the redesign and further development of the program. This version also includes practical checklists for the programmers to use. The guideline consists of three parts; the first section describes usability in general and the potential benefits with usability. The second section describes general guidelines for web based applications. These are to be thought of as rules of thumb for any type of further development and therefore to be applied more or less in specific situations. The third section is specific for Docomotive with examples on how to design specific parts of Docomotive, such as the main application, information windows and tools. There are also descriptions on how to choose new colors and creating new icons. Finally, two checklists are provided for fast checkup of usability issues during development of new functions and improvements of Docomotive. This guideline is based on the analysis and some established usability guidelines (Nielsen, 2000, http://www.usabilitynet.org/, http://www.bunnyfoot.com)

8.1 Usability Explained

8.1.1 What is Usability?

There are many different definitions of usability, but basically any product that has good usability exhibits these 6 characteristics:

- Quick and easy to learn.
- Efficient to use.
- Allows rapid recovery from errors
- Easy to remember
- Enjoyable to use
- Aesthetically pleasing

This sounds obvious and straightforward but unfortunately there is no 'cook-book' approach to ensuring usability. This is because each application has a different business objective and different target users. Therefore each application must be purposefully designed at each stage with the wants, needs and expectations of the users as a priority whilst ensuring that it is delivering the business objectives.

To successfully achieve this, input from real users should be evaluated at every stage during the design process - planning, prototyping, final product testing, revisions and updates. However, it's not too late even for established sites and applications to set about improving usability.

8.1.2 Benefits of Usability

Increased usability benefits both the user and the application provider.

Users:
- Experience satisfaction instead of frustration when interacting with the application.
- Achieve their goals effectively and efficiently.
- Gain confidence and trust in a valuable resource.

Providers save money through:
- Efficient design - employ valuable resources to add value, not frills.
- Reduction of support costs - unusable applications require costly telephone or e-mail support, or user training.
- Increased productivity - intranet users retrieve or post information efficiently; avoids wasting time.
- Eliminate over design - over designed web applications are slow to download and add little value to the user
- Increase accessibility - maximize the potential audience and/or potential users and customers.
- Gain and retain happy loyal users.

8.1.3 How to Achieve Usability
No web application can claim to achieve 'ideal' usability. However, all web applications can reap the benefits of increased usability by adopting a user-centered approach to application design and production. It is best to design for usability from the outset - by allocating time and resources to the planning and development. However, established applications can dramatically improve by identifying and correcting usability problems.

8.2 Usability Guidelines for Web Applications

8.2.1 Introduction

These guidelines are offered to allow developers to make informed decisions about crucial aspects of web application design. They are not intended to be a definitive list or necessarily applicable to every situation - the pros and cons of ignoring each guideline should be carefully evaluated.

8.2.2 Planning and Management

Design for the users - make the whole application user-centric, user-driven. Identify the main purpose of the web application and who the intended primary user is; the wants, the needs and the capabilities of the user. Ask real people at each stage of development. Ensure that the application delivers what it is meant to, effectively and efficiently. Regularly maintain and update the application - avoid link rot, and out of date information. Check that all links work and update or delete as necessary.

Test, Test, Test...:

- Test how real users interact with the application.
- Test each page for validity of HTML.
- Test that each page works in a range of different browsers. (Docomotive is optimized for Internet Explorer)
- Test that each page works for a range of screen resolutions, window sizes, color depths.

8.2.3 Design Concepts

Separate meaning from presentation.

Divorcing content from style uses the web the way it was meant to be used. The content and information structure of each page should be a priority. It is not necessary or even optimal for pages to appear identical for each user, even though it is important for the user to look and feel whereabouts in the application they are and what they can do.

Design for accessibility - enable all users on all devices.

Accessibility is about enabling all users to browse the information provided in the way that they choose - a good usability practice. In addition, there may also be a moral and/or legal obligation to avoid discriminating against users with certain disabilities.

Avoid tenuous metaphor - only use if absolutely appropriate.

Metaphors can aid memory and learning if they are appropriate, e.g. a calculator like interface for performing calculations. However if used for effect rather than function they can confuse and be detrimental to usability.

Aim for rapid speed of delivery.

Speed of delivery is the number one design criterion. Users demand fast page downloads above all else and will not wait for a slow page to download no matter what its content. Human factors research has shown that when interacting with computers response times of less than 0.1 sec are required in order for the user to feel that the system is responding instantaneously, while an upper limit of 1.0 sec is required in order for the user to maintain an uninterrupted flow of thought. (Always provide feedback for users when users have to wait when system is operating.) Anything above 10 sec and users typically lose focus and will turn their attention to other tasks.

All pages should be designed to download as rapidly as possible and ideally within 10 seconds. This means focusing on the content of the page above all else and including graphics and other bandwidth intensive features only if absolutely necessary.
8.2.4 Design - Do's and Don'ts

Use Style sheets. Avoid the deprecated <FONT> tag.
When used correctly style sheets aid usability by providing consistency throughout a site or web application.
Using style sheets rather than inline <FONT> tags also aids maintenance and updating of the site - changes can be made to a single style sheet that have a global effect rather than having to delve into each individual page.
As style sheets are only implemented in the latest browsers care should be taken to ensure that the pages still make sense even when style sheets are turned off.

Avoid too many popping up windows on user’s screens.
People like to be in control of their own machine and many find it highly annoying to have a site or web application spawn all over their screen. Also a new window more often than not obscures what might be useful information in the window below, and more importantly breaks the functionality of the back button (see next guideline).

Do not break the back button.
The back button is the most commonly used navigation method. Users expect to be able to return to previously viewed pages by using it. Popping up new windows or programming automatic redirects etc. break this functionality and greatly annoys and confuses users. Maintain the functionality at all times (or were applicable).

Use Animation when appropriate, but avoid over use.
Animation can often be used to specifically enhance a message or draw attention to an important feature. In many cases, however, animation is used less judiciously and merely annoys users and creates a size overhead. Avoid misusing or overuse of animation.

Use plug-ins only where absolutely appropriate.
Plug-ins technology such as flash can greatly enhance the functionality and appeal of a site or web application, however there is a risk of alienating those users that do not have the plug-in by making the site meaningless or unusable, or by requiring them to spend time downloading the plug-in. Use content requiring plug-ins only if the page absolutely requires it, if used then provide an alternative with equivalent content.

8.2.5 Navigation

Provide a consistent and intuitive Navigation method.
There are many different methods of navigation to choose from. The choice often depends on the purpose and content of the web application. Choose a method that is simple and requires the minimum of learning. Make sure it is consistent across the whole of the site.

Facilitate the rapid retrieval of information by providing a search function, index or map.
Particularly if a web application is large there should be some method provided so that users can quickly find the information or page that they are looking for without having to follow a whole set of predefined steps. Providing shortcuts also facilitates ‘expert users’.

Let users know where they are and where they have been.
Provide a context for each page:
- A meaningful heading.
- Some indication of where the page lies in the application structure.
- A visual cue on a navigation bar or menu.
- Where possible use text links in standard colors so that they change to red/purple to provide feedback about where the user has been.

Make links meaningful and predictive of the destination - DON'T use 'click here'.
Links should be meaningful and if possible meaningful to the extent that even if taken in isolation the user would have a good change of predicting the content of the destination page. Avoid 'click here' and similar variants - give the user more to go on.
Use image links and image maps conservatively.
Linking using images can be used to great effect; however they suffer from several problems:
It may not be immediately obvious what or where the link is.
Image links do not change to show when the link has been visited - breaking a good usability feature.
Text used in images can suffer at different resolutions and become unintelligible - making the link difficult to comprehend.

Avoid links that refer to the current page.
A link should always take the user to another page and not simply reload the exact same information. If the link is part of a menu then change its appearance to provide feedback about the current location.

8.2.6 Page Design

Use high contrast, avoid complicated backgrounds.
For maximum clarity, use high luminance contrast between foreground and background elements. Do not rely on combinations providing color contrast alone as these are generally more difficult to perceive and may cause problems for some users with certain monitors or the color blind. If using backgrounds then make sure they are very subtle and do not interfere with the foreground text or pictures. Also make sure background images are of small size and can be easily and seamlessly tiled to minimize overhead in page download speed.

Left align the majority of text.
Users can read left aligned text much faster and more comfortably than centered or right aligned text. Use effects other than left alignment conservatively to add emphasis.

Use only a limited number of carefully chosen fonts.
Two or three different fonts are probably the limit, any more and the page can appear messy and unstructured. Use fonts consistently to reinforce the relative importance of headings etc. Avoid defining very specific fonts as some users may not have them loaded on their machines. For each font definition provide alternatives and a default. E.g. font-family: Arial, Helvetica, sans-serif.
For smaller text sans-serif fonts may be preferable to aid legibility since serif fonts can become irresolvable due to the relatively low resolution of most monitors. However, many users find serif fonts more pleasant to read. An informed choice must be made.

Do not use all CAPITALS
On average it takes approximately 10% longer to read a block of text composed entirely of uppercase letters. Also it is regarded by some as rude because it is SHOUTING!

Use page space sensibly to maximize visual information.
Grouping, proximity, continuity and uniformity can all aid recognition and meaning. Use the space available on the page to add visual structure by grouping like elements etc.

Aim for scan ability.
Use bulleted and numbered lists instead of large chunks of text
Provide a way for a user to get an impression of the page or section quickly and easily:

Always provide the pixel size for graphics and the width of columns
This allows the browser to reserve space for all of the content quickly. Whole page rendering times will improve and the user will have access to the important text information provided at the top of the page without waiting for all page elements to finish downloading. By the time they are ready for the graphic etc. there is a chance it will have loaded completely.

Use relative parameters to set the size of page elements
Page elements such as table widths and column widths should be specified in percentages if possible to facilitate resolution independent design.
8.2.7 Content

Produce content specifically for the web. Don't simply provide reproductions of documents more suitable for other media, instead adapt old content and produce new content in order to utilize the flexibility and power of the web (see other guidelines in this section).

Prioritize important information - put it near the top. The first thing that loads and the first thing a user sees without scrolling should provide as much information relevant to the context or function of the page. Users can rapidly make an informed decision about whether the page contains what they are looking for.

Prioritize the main points - provide an overview with links to more detail or specialized information. Avoid bombarding users with too much superfluous detail; instead provide links so that those interested in the finer points can view them whilst other users can gain an overview of the topic. This way all the main points can be communicated rapidly. With long detailed pages many users will cease reading before the end of the page and may miss out on important information.

Use plain language - avoid boasting and jargon. Users want the facts explained to them quickly and easily. Overt self promotion or boasting, jargon and bull obscure the facts and damage the reputation of the application.

When appropriate provide alternative copies of pages to facilitate printing. Because of the limited resolution of the majority of monitors (all in fact but the very high end and very expensive ones) text is more uncomfortable and slower to read online than when printed out in hardcopy. Many users will print out important pages to read rather than read them online. Many pages degrade to look terrible when printed out and unnecessary element such as menus, frame content etc. waste valuable space. If appropriate alternative versions of pages should be provided specifically for printing, they can utilize all the typical typographical conventions that make for good looking readable hardcopy which may not (usually are not!) appropriate for online content.

Keep content concise and to the point. Keep the user informed and provide only what users are expecting to find.

Adopt an appropriate tone. Pitch the writing style and the tone of the content appropriately for its intended users. Humor can be a great asset but overuse is a bore and it can obscure meaning.

Ensure any error messages are informative and help resolve the error. All error messages should be in natural language and provide a useful description of the problem and either a means of solving the problem or a method of recovering from the error.

8.3 Design Approach of Docomotive®

The structure of Docomotive is open, which means that all information shown for users with same type of user account is available. All information can therefore be viewed and edited by these users. The only mean of notifying users from editing the same information at the same time, is to set a document in state "engaged". When a document is in state “engaged”, other users are notified by a red color in column with icon “Edit document information”. This is only a notification and does not prevent any other user from viewing and editing information. Whenever a user has opened “Edit document information” other users will find that the icon has changed to a padlock, thus indicating that someone is working with the document information. However, user can always unlock a document. When a user have completed editing document and save this information in database, these changes are not shown to others until an already open page is updated from database. Considering this, it is recommended in further development or additional tools to have in mind that shown information may slightly differ from one user to another when using application at the same time.
On a hierarchical level Docomotive is divided into three categories; base application for searching and viewing document information and build structures, popup windows for editing of information and task planning, and finally a number of tools to support users in their work tasks as well as to secure unified and correct information handling. In this view the three categories should differ in the design regarding attributes and layout. It is important to have this in mind in further development as application grows in size and the number of users increases. A clear distinction between these categories will help users to navigate and recognize where they are and what they can do in different parts of application.

Docomotive should (ideally) have no more than three levels in depth for user to complete their tasks. This means, for example, when user task is to edit document information they may use: (first) main application to locate specific document, (second) edit document information in library card and/or (third) edit information thru tool or popup window in library card. In this way the application design will maintain its simplicity and preserve its main purpose – to find, edit and store information.

The design of Docomotive should strive to suit user tasks since Docomotive is developed to help users do their work tasks in an effective and efficient way. Added tools and functions with complicated structure and information handling will only make users work situation more complicated if not implemented with the user in mind. The graphical user interface (GUI) is what the users sees and interact with, and GUI is therefore a crucial part of which design decisions should be of careful consideration, this to achieve the best possible outcome for both user experience and work results. Even if a new tool or function may have an excellent technical solution, the real benefit comes when users interact with the application in their everyday work. In other words: ask users prior to development and test implemented changes or additional tools and functions with the intended users before changes are considered to be final.

The next three sections are to be used as design guidelines to the different parts of Docomotive to ensure consistency and usability.

### 8.3.1 Main Application

The main application is divided into three frames; top, main and bottom. All frames have a white background color and lies on top of a checked background. The top frame consists of the name of the application and the logotype. The bottom frame shows the colors of the documents' status, e.g. “Removed”, “Temporarily Removed”, “Don’t include yet”, “Non-Alex Doc.”, “Building Block” etc. The color explanation is always displayed so that the users easily can interpret the document status in the document list. Icons to the right at the bottom of the page should only be used for add-ons, such as the glossary application “GLAPP”, printing function and so on.

---

**Figure 1. Main Application**

![Main Application](image)
The main layout principle is to group information and user tasks under a number of tabs. At date of writing these are “My central”, “Search”, “Areas”, “Alex” and “Administration”. If several tasks can be done under each tab, these should also (were applicable) be grouped with a new set of tabs within each tab, such as the search tab (see Figure 1). Try to group related tasks under the same tab and avoid mixing information and worker tasks among different tabs, use crosslink’s only when absolutely necessary.

Check and optimize site for viewing in 1024x768 and above to fit various screen resolutions, although many users at time of writing use 1280x1024 on a 19” screen. Docomotive® should be optimized for Internet Explorer 6.0 and up.

Popup windows
There are two situations when new explorer windows should open; (first) forms for editing and storing information and (second) tools for changing displayed information or editing tools. Every new window must have a title describing what the new window contains and give some indication what user can do (see Figure 2).

![Figure 2. Windows should have a Title and description](image)

Not too many windows should be opened at the same time since it may confuse the user and make navigation harder (see Figure 3). Aim for a maximum of three windows opened at the same time. When a new window is opened adjust place on screen to match rest of system. Check also that window doesn’t cover any information in other windows that may be useful in the opened window, this to prevent unnecessary disturbing adjustment and moving around on screen.
Verify also that the same style is used concerning buttons, checkboxes, radio buttons, icons and drop-down as in rest of application. Remember always to group items that are related to each other.

8.3.2 Document Listing

Listing items in various ways is a main function in Docomotive. Documents, Change Content Forms (CCF), reviews etc. can be sorted and listed in several ways. In general these lists are used to search, sort and edit information regarding specific documents. Since these differ in information type and handling, details (e.g. columns heads) need to be adjusted to present desired and useful data. For example, a list of documents should display information such as title, document number, revisions, and the document place in structure. Relevant links and tools for editing, storing as well as viewing of additional information should also be provided. In a list of Change Content Forms, on the other hand, document title and number for each CCF is also shown, but the rest is CCF specific information such as type, date, comments.
Choose carefully which columns with information to use, and if the same information is presented elsewhere, see to it that they are presented in the same way (see Figure 4). (E.g. Column width and placement, style, text etc.)

![Figure 4. Chosen columns when listing CCF's](image)

Group information or icons that are related to each other and show what can be clicked upon. Additional information of where a link or icon leads should also be provided, preferably with mouse over text (see Figure 5).

![Figure 5. Provide mouse-over information that describes where a link leads to and/or what actions that can be made.](image)

When search engine is running it is important to inform the user that the system is operating. In other words, give feedback so that the user knows what is going on.

Finally, when designing - don’t forget the purpose of showing the specific list and that the idea of Docomotive®, which are to only provide users with the desirable information they need.

8.3.3 Tools

Tools and add-ons should be regarded as supporting functions for users. When developing new tools the developer must have a clear picture of what the main purpose of the tool is, when and how it is supposed to be used and by whom it should be used. This is important for creating a supportive function and not just adding functions that neither add functionality nor help the users in their work. Stick to the purpose and make it easy to use and value will be added to the application and work situation.

The design of new tools and add-ons are dependent on its function and purpose, but there are some basic principles that should be followed. First, in layout: use colors, icons and other preferences as used in main application. Second, use tabs to group information and tasks. Third, try to organize user options and task in the
same window, avoid additional popup windows. Fourth, use title and information what this tool is all about. Fifth, provide help instructions – such as mouse over information and help button (if necessary).

When developing new tools it is also important to consider which information is to be displayed for different users. If the tool is regularly updated and/or information edited (e.g., adding new terms to a glossary) by a user with administrator rights, additional information and tools to do so should be available for this user. However, if user doesn’t have these rights, this should not be shown since it may confuse the user with more and unwanted information. Concerning this, information could be presented in different ways to fit user tasks and intentions. For example, a glossary is used by a technical writer to search spelling and explanation of words and phrases. Information for which builds to use this spelling is also of interest. So a search function to support this task should be provided and the information shown should be arranged in the best possible way to visualize this. An administrator, on the other hand, may need additional information for sorting, editing, saving and connecting forbidden terms with allowed terms. Don’t try to squeeze in to much information to fit users with different tasks in the same window.

8.4 Colors

The use of colors can be a very efficient way of enchanting usability and information visualization. However, an overuse of color can impose an opposite effect. Too many colors will confuse users about the meaning of colors and color coding, as well as making information hard to find and read. Instead of showing meaningful relationships, inappropriately colored fields mislead users into searching for relationships that do not exist. Therefore colors should be used carefully and with (ideally) a consistent and well thought-out color coding.

When using colors with text, either colored text or text on colored background, the combination is vital to support good readability. For example, use a solid color as background to text (see Figure 7); don’t use a pattern since this can make the text “floating” in its appearance (see Figure 6). Use illumination and saturation to enhance and separate color fields instead of hue. For example, when a block/area of information is to be brought out from the rest; use a brighter color on a darker background.

![Figure 6. Don’t use text on patterned background](image)

![Figure 7. A brighter non-patterned background is used bring out the area of interest and for support of good readability.](image)
(“Removed”, “don’t include yet” etc.) (Figure 8) and another set of colors that represent button actions, such as save, cancel, delete etc. In general: use colors when they have a meaning and be restrictive otherwise. And if you use color codes make sure that they are consistent throughout in the entire application.

![Color codes for document states](image)

Figure 8. These colors are a set of colors that can be used for color coding of a document's state.

8.5 Icons

There are always situations when the choice has to be made whether to use an icon or stay in a textual design. Does an icon explain the purpose better and thus saving space, or maybe text is needed since an appropriate metaphor can not be found? The style on rest of application has to be in mind when deciding this. However, if you choose to use an icon there are some guidelines that are to be followed. (Shneiderman, 1998)

- Represent the object or action in a familiar and recognizable manner
- Limit the number of different icons
- Make the icon stand out from its background
- Consider three-dimensional icons; they are eye-catching, but can also be disturbing
- Ensure that a single selected icon is clearly visible when surrounded by unselected icons
- Make each icon distinctive from every other icon
- Ensure the harmoniousness of each icon as a member of a family of icons
- Add detailed information, such as shading to show size of a file (larger shadow indicates larger file), thickness to show breadth of a directory folder (thicker means more files inside), color to show the age of a document (older might be more yellow or greyer), or animation to show how much of a document has been printed.

Icons are very useful in creating relevant options in specific situations. However, there is also the decision to be made if an icon, link or button is to be used, depending on the intended function. But if the decision falls on an icon follow the guidelines above for the creation of the icon. As for colors in the preceding section, icons have to be grouped in categories with a color scheme that is common for all icons in Docomotive. For example, one category for editing and changing information related to a specific document and a second category for additional functions or tools for general use (Figure 9).

![Icon examples](image)

Figure 9. Grouping icons with similar purpose: Links to show information, icons for tools and state, respectively.
Use few details when designing the icon and pick colors that are used in other icons. If additional colors are added, see to it that these match the rest.

Shadows can be used since it makes the icon “stand out” from the screen, hence implying to be clicked upon.

If text is used under the icon, be as precise and straight to the point as possible. Additional information should be used in mouse-over popup. However, when writing this text it is important not to repeat the already said information. Ideally, this text should inform user where the link goes and what tasks that can be done.

When designing an icon, unless your icon covers the entire canvas, you’ll want to make the background transparent. This will allow the canvas of the icon to be transparent and enable the background to show through. Always check if icon goes well with existing and possible background colors.

Finally, once again, make sure that the new icon goes in line with existing icons in the same group concerning colors and style.

8.6 Checklists

There are two checklists (appendix. A, B) provided in the design guideline. These are to be seen as a quick guide for applying usability aspects in Docomotive and are based on some established usability principles. However, for more specific information and layout styles developers are recommended to use the rest of design guideline. The first checklist consists of 20 checkpoints that developers can use to apply usability aspects during development. Depending on what new functions and tools are developed some of them will apply and others not. But in general they all should be checked before implementation is to be considered final. The same applies to the second checklist albeit this is more directed to usability for programmers concerning support and further development independent of who and how many programmers that are involved.

9 DISCUSSION

Docomotive as an application is not that complicated in structure. However, when several developers have worked with add-ons and new functions, different styles in color, layout and language have been used. A problem with Docomotive as an application for keeping track of document information is; that even though most tools and information needed is available, users have trouble finding it and thus spend a lot of time trying to find their own solutions to complete the tasks. “I know that all the necessary information is there, but how do I extract what I need from the system?” (User)

The choice of methods (Heuristic evaluation, observation with think-aloud technique, survey) were well suited for the purpose of finding many usability issues concerning Docomotive. Even if some problems were discovered in more than one method, the overall amounts of problems discovered from each method were high. The methods and the design of the methods were chosen to complement each other and thus both validate discovered problems as well as covering different aspects of usability problems found. For example, subjective reports of how the users are supposed to use Docomotive may not always concur with their actual interaction. Such difference can be found by combining two methods such as semi-structured surveys and observations with think aloud technique. On top of this, such as which ways users take to solve a task or how they scan the GUI and click around objects can be discovered in the observation, but not with surveys.

The sample from population used in all methods was only users of Docomotive. The reason not to involve a control group outside this sample was that the present GUI was very hard to understand and use, even for the authors of this report it took considerable time to understand and use Docomotive. Therefore, the choice of methods and method design was selected due to the fact that all interview persons and participants in the observation were active users or had experience with Docomotive. However, if the goal would have been purely implementation, the heuristic evaluation itself would have found many problems within the GUI. Many basic usability issues were broken and correcting these could easily make a significant impact on increasing the usability of Docomotive.

After completing the analysis and evaluation of Docomotive the work was directed towards establishing some design guidelines, and thus the implementation part became secondary. It was secondary in the meaning of only implementing the most crucial aspects for correcting usability problems in the most used parts of Docomotive. It was also important to implement these parts to make examples as reference for further development. The rest were to be supplied as a to-do list for SID developers. However, since the existing code was very hard to get into (several programmers with different styles of commenting, if comments were made at all) a considerable amount
of time had to be spent, relatively, for the accomplished implementations. Time spent on finding suitable metaphors for icons and drawing were also a time-consuming part of the design process.

The latter evaluation showed some complications between developers and users with different accounts and responsibilities. For example, some technical writers liked the new design of page “Areas”, a builder did not. The page “areas” are more used by builders and their opinion may therefore be of higher interest, even if the technical writers are in greater numbers.

The time schedule worked well even though it can be hard to estimate time spending for different parts of the evaluation and design process. All parts of the process were conducted as planned, although a one week delay concurred as a result of other implications during Christmas and New Year holidays.

In the later stage of the design process a potential limiting factor occurred, the concept of expert blindness. As the knowledge of the system and the documentation process increased, a possible narrowed view of new design possibilities and new solutions may have affected the design decisions made. With this awareness, efforts were made in trying to limit the impact of expert blindness.

Conducting a usability analysis at a company has several aspects to be considered which have an impact on the design process. Time and costs for different choice of methods as well as the availability of users to perform user testing have to be adapted to the surrounding circumstances. In this case there were also no updated documentation of how Docomotive works and how it’s supposed to be used. Docomotive is under constant development and thus the upkeep with documentation falls behind. There were also several old parts and functions in the application that are not in use anymore, or are in use in ongoing projects but are to be outfaced when the project ends. These parts are not considered in the design proposals but are still used in their existent form until they expires. During this time the design of Docomotive will not be consistent. Hopefully, with help of this master thesis and the design guideline, Docomotive will improve usability and consistency in further development.
References


Web Resources

http://www.bunnyfoot.com, Feb 2006


APPENDIX A – Usability Checklist

This checklist is provided in order to facilitate a quick assessment of a web application conformance to some established usability principles.

1. Add-ons and new design was designed with the aid of user input from the outset and major revisions have also re-evaluated the wants, needs and expectations of the target users.

2. Add-ons and new design is meaningful and functional at a variety of screen resolutions and window sizes.

3. The application utilizes a simple and intuitive navigation method. The pages are consistent so that user’s can predict where important page element will occur.

4. Avoid tenuous use of metaphor unless absolutely appropriate. Where icons, pictures, or special terminology are used to represent actions, they are intuitive, consistent and well known.

5. Each page loads quickly. If there is a substantial delay in loading the whole page (greater than 10 seconds) the most informative and important items become visible and are meaningful while the rest of the page loads.

6. Frames are not used. If they are, equivalent non-frames content is provided via the <NOFRAMES> tag.

7. All pictures have meaningful alternative descriptions provided via the <ALT> tag and <LONGDESC> tag where necessary.

8. All text links are meaningful and predictive of their destinations.

9. Animation is used only where it supports the meaning of functionality of the page.

10. Page titles are concise and meaningful

11. Where link titles are used they provide valid extra information and don't simply echo the link itself.

12. The location of the page being viewed is indicated clearly.

13. Links on a current page do not simply reload the page itself. (No self-referential links.)

14. Windows are only popped up on a user’s screen where necessary, and if they are, no more than three windows are opened at the same time to solve the same task.

15. Back button functionality is, where appropriate, preserved.

16. Error messages are written in clear, understandable language and indicate how to resolve the problem.

17. Where appropriate a good search facility is provided - this uses a sensible algorithm and if possible supports spelling mistakes.

18. Complicated backgrounds are avoided. Where used backgrounds do not impair the contrast between foreground and background.

19. Prove feedback to users whenever page takes long time to load, email is sent or information is saved.

20. Differentiate on what is “clickable” and what not. Links, icons and button that are “clickable” must indicate this. Make icons and buttons stand out from its background and use links related actions. (underline, color, cursor)
APPENDIX B – Programming Checklist

This checklist is provided in order to establish some code writing principles. This will help updating and changing information in the future for all programming tasks.

1. **Use style sheets** and not inline `<FONT>` tag.

2. **Comment code in a consistent way** throughout whole application.

3. **Keep code clean and tidy.** If parts of code are marked as comment and new code is written, **make comments** or delete code that is not used.

4. **Intend code** for good readability.

5. When writing code for listing posts, make them **as time-saving as possible.** Even the slightest time saved when executed each line can make a significant difference in loading time with long lists, thus keeping speed of delivery as fast as possible for the user.

6. **Don’t mix new and old files in the same directory.** Overtime the number of files grows and thus making traceability in history hard. Use name, date and by whom for backup and storage of older versions.
Planeringsguide

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


Introduktion

Förstudie

Metodval

Användarstudie

Producenter Design och Konceptförslag

Implementering

Utvärdering

Opposition och Presentation

Slutgiltig Revidering av Rapport