Design & Development of Web-based Information Systems for Port Operations.

Caroline Olsson

Master's Thesis

2004 Department of Design Sciences
LUND UNIVERSITY
ISRN: LUTMDN/TMAT-5071-SE

Sölvesborgs Stuveri & Hamn AB

Abstract

The objective of this thesis was to make a study on how to develop self instructed web applications for Sölvesborgs Stuveri & Hamn AB, SSHAB, that will be user friendly and that will optimise the users' interaction with them. This, so the applications will become an effective information source for the employees and the customers to SSHAB. With this study as a starting point, the applications will be designed and implemented.

From several brainstorming sessions and early user studies, requirements for the different applications were gathered and together they formed the usability goals and user experience goals for the applications. These goals were kept in mind during the whole design process and the applications were constantly tested on the intended users to get feedback about how they perceived them, in order to see if the applications were moving closer or farther away from these goals. By doing this, the generated designs were constantly improved during the whole design process so that they in the end were going to be usable and accepted by the users. As a help, a thorough literature study was made which was used, together with the established goals, to guide the design process.

The result was three web applications; a homepage, a logistic application and an application where the information about the arriving vessels is going to be updated. These three applications are together going to work both as an important source of information but also to facilitate the information torrent between the steel suppliers in Europe, SSHAB and Volvo.

Sammanfattning

Målsättningen med detta examensarbete var att göra en studie om hur självinstruerande webbapplikationer för Sölvesborgs Stuveri & Hamn AB, SSHAB, skulle kunna utvecklas för att bli användarvänliga och så att användarnas interaktioner med dem optimeras. Detta för att applikationerna då skulle bli en effektiv informationskälla för de anställda samt kunder till SSHAB. Denna studie användes sedan som grund för att designa och implementera applikationerna.

Med hjälp av ett flertal brainstormingsessioner och inledande användarstudier, kunde kriterier för de olika applikationerna sättas upp, vilka tillsammans formade användbarhets mål och användarupplevelse mål för applikationerna. Dessa mål fanns i åtanke under hela designprocessen och applikationerna testades kontinuerligt på de avsedda användarna för att få feedback om hur de uppfattade dem och för att kunna se om applikationerna rörde sig närmare eller längre ifrån dessa mål. Genom att göra detta, förbättrades designerna hela tiden under designprocessens gång så att de till slut ska vara användbara och accepterade av de blivande användarna. Som en hjälp, gjordes en omfattande litteraturstudie vilken användes tillsammans med de uppsatta målen för att styra designprocessen.

Resultatet blev tre webbapplikationer; en hemsida, en logistik applikation och en applikation där information om de anländande fartygen till hamnen ska uppdateras. Dessa tre applikationer ska tillsammans fungera både som en viktig informationskälla men även för att underlätta informationsflödet mellan ståleverantörerna i Europa, SSHAB och Volvo.

Preface

This Master's Thesis, "Design & Development of Web-based Information Systems for Port Operations" has been prepared as a compulsory part of the Master's education in Computer Science at Lund Institute of Technology.

The work has been carried out at Sölvesborgs Stuveri & Hamn AB in Sölvesborg, Blekinge in co-operation with the Department of Design Sciences, the Division of Ergonomics and Aerosol Technology at Lund Institute of Technology, during 2003-2004.

I would like to thank the people that have helped me throughout the development of this thesis and the persons I especially would like to thank are…

Jan-Erik Olsson, my supervisor at SSHAB, for all the support concerning practical matters and for always having the time to answer my questions.

Jan-Olof Krantz, my other supervisor at SSHAB, for contributing with useful ideas and for taking time to help me during the work.

Joakim Eriksson, my supervisor at Lund Institute of Technology who has contributed with his expertise of human – computer interaction and for helping me to structure the work with the thesis.

All the participants of the usability tests.

Thank you!

Caroline Olsson

Lund, October 2004

Table of contents

1 Introduction

1.1 Background

Sölvesborgs Stuveri & Hamn AB, SSHAB, is one of Sweden's fifty-two ports and is located in Sölvesborg, Blekinge. SSHAB have forty employees and have a turnover of fifty-five million Swedish crones per year. Out of these fifty-five million crones, approximately twenty million crones apply to Volvo's activity, i.e. loading, unloading, storing and transporting coils[1] for the automotive industry. The remaining part, approximate thirty-five million crones, applies to loads and unloads of pulpwood and storage handling of paper products.

This master's thesis will only consider the Volvo activity where the suppliers of steel all around Europe send coils via vessels, trucks and rail to SSHAB who then stores these until they get called by Volvo.

The information transference is managed by EDI, which stands for Electronic Data Interchange and mean that the information is transferred electronically with help of standardized messages. These types of messages are used by the steel suppliers around Europe to send information about the coils they are sending to SSHAB and Volvo.

The information flow goes like this; Volvo puts together a delivery plan to the suppliers of the material they are going to need within a certain period of time and from this delivery plan the suppliers deliver the goods to SSHAB who then stores them. Volvo does not have any storage possibilities on their own so the steel they get delivered, via trucks and trains from SSHAB, goes directly into production. When the goods are sent from the suppliers, SSHAB are being notified electronically by an EDI message.

When the coils arrive at SSHAB, their bar codes are scanned with help of a hand held computer and are then put into stock. When this is done an EDI message is sent, both to the suppliers and to Volvo, to notify them that the goods have arrived. Volvo makes an electronic order of the coils they need the following twenty four hours and the material is pre planned by SSHAB. When the coils leave the warehouse at SSHAB, Volvo is notified electronically by an EDI message that their coils are on their way.

1.2 Problem

For this system to work a logistic application had to be developed where the suppliers, the employees at SSHAB and at Volvo will be able to see their orders, how the goods are being delivered, which coils that are in stock, block faulty steel etc. This is especially important for Volvo since if the coils they have ordered are not in stock at SSHAB they have to re-plan their whole production. These functions had been handled by the employees at SSHAB exclusively before but are now going to be handled as a web service via SSHAB's new homepage, i.e. all paper work are going to be abolished where possible and the information torrent is going to be done electronically and via SSHAB's homepage.

The main functionality for this logistic application was already working at the beginning of this project. However, no thought had been put into the design of it to make it easier for the

[1] High quality steel in the form of rolls that each weighs approximately 15-25 tonnes.

users to use and to make it look appealing. See Appendix A for a picture of what this application looked like at the start of this project.

As mentioned above this logistic application is going to be accessed through SSHAB's homepage which has to be developed. At the start of this project SSHAB already had an existing homepage but this was not used as a base for the new one since it had too many faults and were too far behind the competitors' homepages. See Appendix B for a picture of the old homepage. This old homepage was created three years ago by two of the employees at SSHAB, without any consideration to the theory of user centered design and it has not been updated since then, hence a totally new design is needed. The new homepage has to be selling and competitive since it is, after all, going to be SSHAB's face towards their existing and future customers.

1.3 Objective

The objective of this thesis was to make a study on how to develop self instructed web applications for SSHAB that are going to be user friendly and that will optimise the users' interaction with them so they become an effective information source for the customers, employees at SSHAB etc. With this study as a starting point the applications will be designed and implemented.

It is important that the homepage, which is the starting point for the logistic application, is designed to support the users so they, fast and easy, can understand how to interact with it. It is also important that it is attractive and easy to use. The functions on the applications need to be intuitive and thereby easy to use or else they will cause problems and confusion for the users and probably not be used at all.

As mentioned, the logistic application for Volvo and suppliers had partly been developed by another person when this project was started however, no aspects of making the application user friendly had been taken into consideration. So the objective of this application will involve making it usable to the users as well as making it aesthetically pleasing. In addition, the application is going to be directly accessible via the homepage.

When it comes to the safety for the applications, it needs to be seen to so that no unauthorised persons can get hold of important information and that customers only can access the information that concern them.

1.4 Delimitations

A chosen delimitation for this project was that the applications mainly are developed for companies and customers to SSHAB, not so much for private persons. Hence, the assumptions that all the users of the applications are using broadband, not a modem, were made. This will have a great importance since the homepage contains a lot of images that would have taken forever to download if a modem was going to be used. The ambition, though, has been to design the applications so that as many persons as possible can take advantage of them.

Another delimitation made for this project was that the usability tests were just performed on the employees at SSHAB, not on their customers, who also are going to be users of the applications. This was done due to the reason to keep the costs and the development time

down. However, since the employees at SSHAB are going to use the applications as well, they are representative users of them, hence the results from the usability tests will be able to be used to see if the applications are user friendly and if they meet the needs and expectation of the intended users.

2. Theory on usability engineering

2.1 Background

In the early days of computing[2], only high skilled technical people used computers since they were very ungainly, expensive and complex to interact with. This because they had been designed by specialists for specialists [1]. However, during the past 30 years, technology has advanced to such an extent that almost everyone, young or old, able or disabled, skilled or unskilled, come in contact with computers, in one way or another, regardless of if they want to or not. This is due to the reason that today's computers can appear in every size and shape and a lot of them are not even thought of as being computers [2]. They are everywhere; at public rooms, at workplaces, at meeting places but also in people's homes and during their spare time. In a lot of ways, computers shape people's lives hence, it is important to support their interaction with them so that they are effective and easy to use [3].

Today[3], a lot of products that require interaction from the users, to carry out their tasks have not necessarily been designed with the users in mind. So, although the product may work effectively technically, the users may not be able to interact with it because it is to complex [4]. In order to minimize the problems with technology and to make technology work for the users, artifacts[4] must be designed with the intended users in mind. That is, the technology should be designed to meet the needs of the users. Users must not be required to change to meet the needs of the technology [5]. This is where human-computer interaction, HCI, and interaction design comes into the picture. In these two areas, the users' concerns direct the development of the product rather than the technical concerns and the emphasis of an understanding, and indeed the active involvement of the people who will be affected by the design, is central [6].

2.2 Human-Computer Interaction

HCI was adopted in the mid 1980s as a mean of describing a new field of study, concerning all aspects that relate to the interaction between users and computers, not just the design of the interface[5] [7]. The initial vision of HCI was to bring cognitive science[6] theories and methods into software development and its primary goal was to make computers easier to use. It was hoped that these methods and theories were going to provide substantive guidance at a very early stage of the software development process [8]. They were going to help increase the understanding of technology and its effects, to discover what impact computers are having on people's communication with each other, their productivity, job satisfaction and their lives. With this knowledge the quality of the interaction between humans and computer systems is likely to be enhanced [2].

Norman states that we must create cognitive artifacts that will embrace our strengths and replace our weaknesses. By this he means that if technology is designed to match the cognitive characteristics of the users, i.e. their needs, abilities, and preferences, the users will

[2] The first computers appeared on the commercial scene in the 1950s [7].
[3] 2004.
[4] Computer systems. http://www.eat.lth.se/Kurs/Material/MAM061
[5] Those aspects of the system that the user comes in contact with [7].
[6] The study of how people carry out their everyday tasks and how they cope with different confusing stimulus [7].

be happier about using the technology, will work more productively and will have or cause fewer accidents [5]. The goal is to create a user-centered interface, which will provide intelligent, understandable, tools that bridge the gap between the users and the systems [6]. Technology used to have to fit people's bodies, now technology must fit people's minds [5].

There are numerous different definitions of HCI. Two definitions that together captures the essentials of HCI are the one by Helander et al. [2]; *"in human-computer interaction knowledge of the capabilities and limitations of the human operator is used for the design of systems, software, tasks, tools, environments, and organizations. The purpose is generally to improve productivity while providing a safe, comfortable and satisfying experience for the operator"* and the one by Preece et al. [7] *"HCI is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them"*.

The main objectives of HCI is to advocate a user-centered design i.e. primarily focus on how to identify user's needs to be able to bring usability into the design process, so that the products become usable and understandable to the users [9]. Usable means that the product is easy, effective and enjoyable to use - from the user's perspectives [4]. The concept of usability will be discusses further in Section 2.4.1.

HCI specialists strive to develop techniques to help designers ensure that computer systems are usable to the user to help them achieve their goals [4]. This requires a deeper understanding of the different factors that determine how the users interact effectively with the computer system. To be able to obtain this understanding, HCI specialists have been forced to gather knowledge and methods from a lot of other disciplines, see Figure 1 [7].

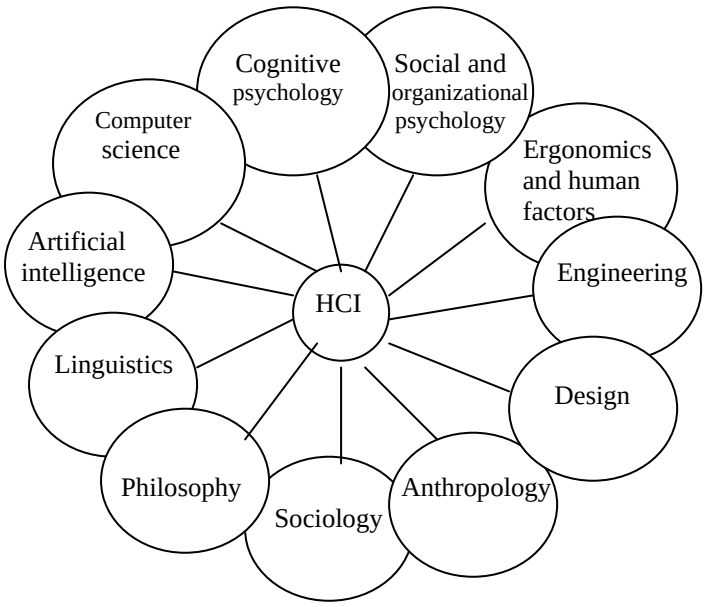


Figure 1. The disciplines that contribute to HCI, author's interpretation from Preece et al. [7].

In these techniques HCI specialists provide different methods with crucial steps to take to be able to produce a usable product. However, the methods are just there to remind the developer of things to attend to and activities to undertake during the design process. There is no such

thing as a "best design" – it depends. There is always a risk that somebody will produce an unusable system even though the methods were followed precisely. Every design proposal has its benefits and shortcomings. Designers need to learn how to weigh-up these pros and cons and be prepared to make trade-offs. To be able to do this in a correct way, designers can not rely totally on theory, they need skills [10].

A way to acquire these skills, a way that is central in HCI, is to work closely with the users during the whole design process, from analysis through design, evaluation, and maintenance. It is important to take into account where the product is going to be used and who is going to use it. A lack of this user-centered focus often results in a product that does not meet the needs or expectations of its intended users [4].

2.3 The design process

Design is to create something new, hence each design process is unique [3]. Ben Shneiderman [11] describes design as: "[*blending*] *a thorough knowledge of technical feasibility with a mystical esthetic sense of what attracts users*". In other words, in order to design the designer has to know what can be done along with the ability to make it look nice too. Getting a design right is not about being complicated, it is about understanding exactly what the users want to do and catering for just those needs.

To be able to get an overall view of the development process that is involved when designing an artefact, a lifecycle model[7] can be used. This way targets can be set, progress can be tracked, and so on. A lifecycle model is just intended as an abstraction, so if it is going to be put into practise details that are specific to the circumstances will need to be added [4].

Different fields have developed different types of lifecycle models e.g. software engineering has the waterfall, the spiral and the rapid applications development, RAD, model and HCI has the star and the usability engineering lifecycle model. Based on these ideas Preece el al. [4] have developed a model showing their view of what a lifecycle model for interaction design might look like, see Figure 2.

This is the model that is going to set the grounds for this thesis and project.

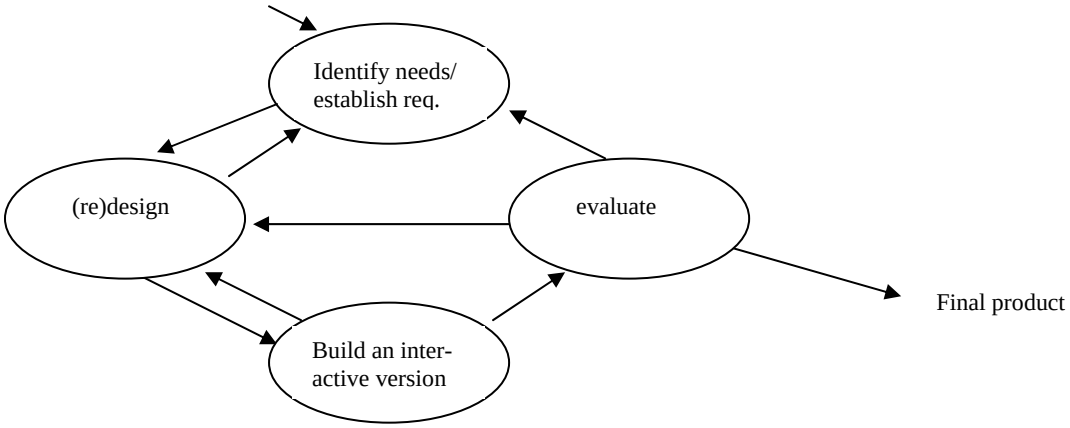


Figure 2. A lifecycle model for interaction design showing how the activities of interaction design are related, author's interpretation from Preece et al. [4].

[7] A simplified model of reality [4].

This model encourages a user focus and incorporates iteration throughout the whole process. It starts with identifying needs and requirements. Then, in an attempt to meet these, some alternative designs are generated and interactive versions are developed and evaluated. Based on the feedback gathered from the evaluations, it might be necessary to return to the first step, identifying needs or refining requirements, or go straight into redesigning. The alternative designs can follow this iterative cycle in parallel, or one at a time. A central theme in interaction design is that design and evaluation are interleaving, highly iterative processes [4].

Iterating between these stages is essential and as Shneiderman [12] stated *"design intrinsically involves the discovery of new goals"* and as Helander et al. [2] claims *"nobody can get it right the first time"*.

By iterating through this cycle, the designer will move from just having a rough initial idea to the finished product in an evolutionary way so that the final product will meet the usability criteria that were decided upon at the beginning of the project. The number of iterations through the cycle may vary from project to project [4]. Often, without an iterative user-designer interaction, there will be little relation between what the designer build and what the intended users are likely to ever use [2].

The model shown above includes the four activities and the three key characteristics of the interaction design process. The four activities are:

- Identify needs and establish requirements.
- Develop alternative designs that will meet those requirements.
- Building interactive versions of the designs so that they can be communicated and assessed.
- Evaluating what is being built throughout the process, i.e., measuring their acceptability of the designs.

And the three key characteristics are:

- Users should be involved through the development of the project.
- Specific usability and user experience goals should be identified, clearly documented and agreed upon at the beginning of the project.
- Iteration through the four activities is inevitable [4].

These will be investigated further in Section 2.4.

As stated above, the lifecycle model showed in Figure 2 is a lifecycle model for interaction design, which is defined by Preece et al. [4] as *"designing interactive products to support people in their everyday and working lives*". They also state that interaction design is about *"creating user experiences that enhance and extend the way people work, communicate and interact"*.

Preece et al. [4] view interaction design as fundamental to all fields, disciplines, and approaches that are concerned with the research and the design of computer-based systems for people. They state that interaction design is *"concerned with a broader scope of issues, topics, and paradigms than has traditionally been the scope of human-computer interaction"*. They see HCI as an interdisciplinary field to interaction design, see Figure 3.

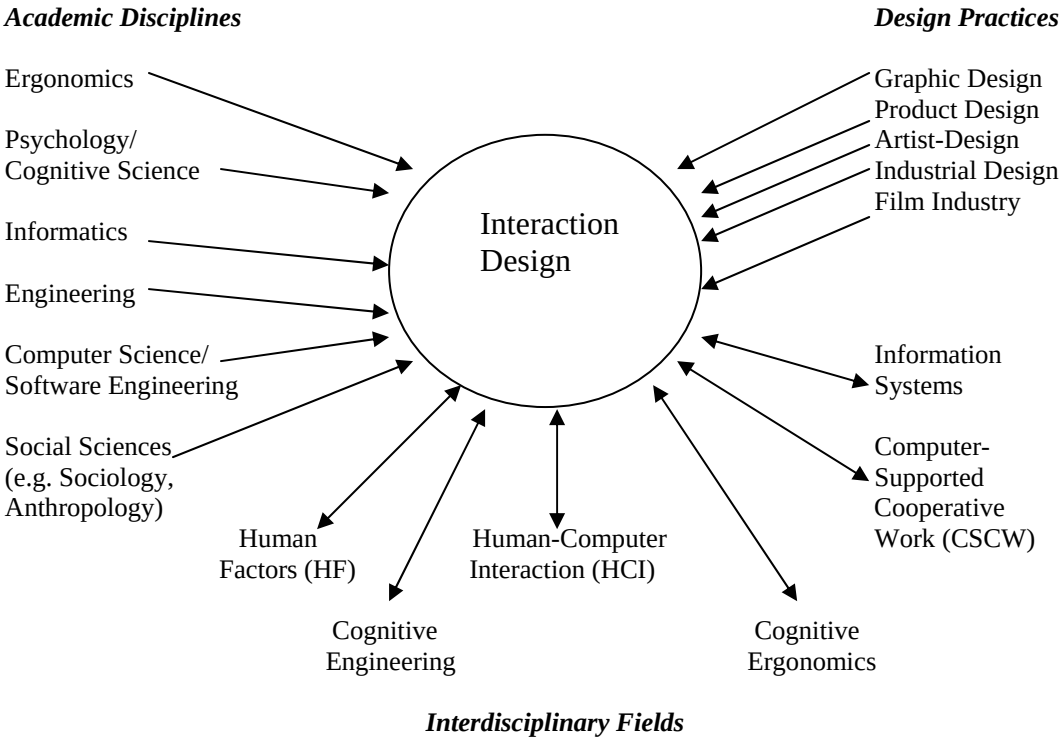


Figure 3. Relationship among contributing academic disciplines, design practices and interdisciplinary fields that are concerned with interaction design, author's interpretation from Preece et al. [4].

2.4 The process of interaction design

As mentioned above, the process of interaction design essentially involves four basic activities; identifying needs and establishing requirements, developing alternative designs that will meet those requirements, building interactive versions of the designs so that they can be communicated and assessed, evaluating what is being built throughout the process, i.e. measuring the acceptability of the design [4]. Each of these will be explained in more detail below.

Keep in mind though, that even if these activities are going to be investigated separately in this thesis, it is intended that they are going to inform one another and to be repeated. Iteration allows for the design to be refined with help of feedback. As users and designers start to discuss requirements, needs and objectives, different insights into what is needed and what is feasible will emerge. Iteration is inevitable because designers never get the solution right the first time [4].

2.4.1 Identifying needs and establishing requirements

A main objective of interaction design, and one of the biggest problems when developing a human computer system, is to optimise the interaction users have with it [13]. This requires the system to support the users' needs, match their wants, and extend their capabilities hence the activity of finding out these things, identifying these needs, wants and establishing requirements, is fundamental [4].

Norman [5] claims that when artifacts are designed to meet the needs of the users and those artifacts take advantage of the affordances[8] of the technology, the person and the artifact become an effective system. The person and the artifact together become smarter than the person or the artifact would be on their own.

The identifying needs and establishing requirements activity should be done either before starting to design the system or at a very early stage after obtaining some general ideas about it [2]. However, since the design process is highly iterative, this activity may need to be revisited later on in the process as well. The first step to take is, as stated by one of the key characteristics mentioned above, to specify the usability goals and user experience goals, which will be explained further in the goal analysis in Section 2.4.1. These needs will then form the basis of the product's requirements which specifies what the intended product should do and how it should perform [4]. Only when this activity is performed, the best system for a particular task and a particular user working in a particular environment can be produced [13].

A product's requirements should be as specific, unambiguous, and clear as possible and they can greatly help designers determine which solutions to choose among the many different design alternatives available and how to develop and test these further. If the requirements are wrong the system will be wrong and, as Preece et al. [4] states, the *"product will at best be ignored and at worst be despised by the users, and will cause grief and lost productivity"*. A good rule to follow is that if the users can not use the system, it does not work.

However, requirements are not always that easy to identify. The customers that ordered the system may not know what they need, or they may have an initial set of requirements but these still need to be analysed because the customers most certainly have not explored them in sufficient detail and are most likely to have missed a few essential requirements for the system to be able to be usable. The customers may not know what is possible [4]. In Section 2.4.1 a couple of different types of requirements and a couple of different methods to help collect these are introduced.

As mentioned, the development of a set of requirements is an iterative process of evolution and negotiation and it needs to be managed and controlled carefully. There is no specific time limitation for how many iterations that are needed. In practice, the users will interact with the design and the requirements will evolve from this. As shown in the lifecycle model in Figure 2 in Section 2.3, the activity itself will be revisited repeatedly [4].

Different kinds of requirements

Preece et al. [4] have illustrated the variety of requirements that need to be captured in order to be able to produce a usable system and in order to compile the usability specification. These are the user requirements; that capture the characteristics of the intended user group, the functional requirements; that capture what the product should do, the data requirements; that capture the type, volatility, accuracy, size, amount, persistence, and value of the amounts of the required data, the environmental requirements; that refers to the circumstances in which the interactive product will be operating, and the usability requirements; that capture the usability goals and associated measures for a particular product. The designer must strive to balance all of these, which often are in conflict, to get a usable system. Four of these will be discussed further below.

[8] An aspect of an object which makes it obvious how the object is to be used [7].

Since an important part of determining the requirements activity is gathering data, different kinds of data gathering techniques will be described in Section 2.4.1.

Once the early data collection and preliminary requirements are established, more detailed design and early development can begin, but before a specific set of requirements will evolve, this requirements activity may have to iterate a number of times [12]. A deeper understanding of the requirements will emerge the more interpretation and analysis techniques that are applied. This means that the requirements descriptions will expand and be clarified [4]. The requirements definition is then used as the basis for defining the language of interaction between the computer and the users [1].

User requirements

The collection of attributes for "typical users" is called user profiling. It is important to determine these attributes, i.e. determine what user group the product will target and to capture the characteristics of that user group. It is a simple idea, but a difficult and an often-undervalued goal [4].

To be able to produce a useful application there must be a concrete understanding of who the users of a product are going to be, what they want to be able to do with it, what level of expertise they have, what their needs are etc. [13]. A main reason for getting a better understanding of users is that different users have different needs; they learn, think, and solve problems in very different ways. Hence, the interactive product needs to be designed accordingly [12]. For example novice users will require that the interaction is backed up with clear information and experts will require a flexible interaction, and as Tognazzini [14] stated in his book "*design with only a single user in mind, and you will find that only a single user can use your program*".

However, if a product is aimed to satisfy everybody's needs and requirements there is a risk that no one will be contented, hence design tradeoffs are required [2]. As Alan Cooper [15] states

"The broader target you aim for, the more certainty you have of missing the bull's eye. If you want to achieve a product satisfaction level of 50%, you cannot do it by making a large population 50% happy with your product. You can only accomplish it by singling out 50% of the people, and striving to make them 100% happy."

As one of the key characteristics of interaction designed mentioned in Section 2.3, users should be involved throughout the whole development of the project. It may be tempting for designers to simply design whatever they like and what they think the users want, but then there will be no guarantee that their ideas will coincide with those of the target user group [4]. Designers are not typical users, even though they may think of themselves as ones, hence they should never use themselves as sufficient testing of the usability of the product [16].

It is crucial that representative users from the real target group are approached because no matter how well the designers think they know their subject and their audience, the things people tell about their expectations and their priorities will make the product much better. It is a near certainty that costly mistakes can be avoided, and a better reception from the users can be assured, if the time to find out what the users want and need is taken before the

development of the system begins [17]. As Faulkner [13] states *"the failure to consult the end-user can have devastating effects on the developed product"*.

Functional requirements

In order to make a software product usable, designers need a better understanding of the tasks the users will be performing or will want to perform with it in order to reach their goals [13]. This knowledge will then be used to influence the design. So after carefully drawing the user profile, the tasks must be identified before the design can proceed [12].

This activity is called a task analysis and is not simply a question of asking people what they need and then supply it, since users do not necessarily know what they need or what is possible. Task analysis concerns what tasks the users will have to do in order to reach their goals [7]. It should give an understanding of the characteristics and capabilities of the users, what they are trying to achieve, how they achieve it currently, and whether they would achieve their goals more effectively if they were supported differently [4]. To be able to get this understanding of the tasks, the users must be observed directly in the field, which will be discussed further in the next section [7].

Task analysis will help designers understand the details of the task, and once understood, a series of design and evaluation steps follow, including detailed analysis of the visual display, Section 2.4.2, prototyping, Section 2.4.3, and usability testing, Section 2.4.4. The design will then iterate until it meets present target criteria [4].

Environmental requirements

As mentioned above, in order to get a deeper understanding of the environmental requirements and the tasks that are going to be performed with the new system, the users must be observed directly in the field. This is due to the fact that the intended users and the environment, in which the finished system is going to be used, are full of variables. These variables can not always be predicted and can certainly not be controlled, but by taking a lot of these into account while designing the product, it can be designed to work well under a range of probable conditions [17].

As Tognazzini [14] states *"the designer must know the subject he is presenting, and he must know the limits and opportunities of software technology. He also must immerse himself in his user's world"*. Keller [1] adds that the designer needs to *"watch him, study him, interact with him, learn to understand how he thinks and why he does what he does"*. Tognazzini [14] continues *"only then can he see neglected possibilities, unworkable requirements, and the myriad details from which he can form a solution that has conceptual integrity"*.

This activity is called a field analysis, and it is conducted in the user's environment, where the product is to be used, over a longer period of time to get an increasing understanding about what users do naturally. It gives the developer a much better idea about the context in which the system will be operating and it is valuable for confirming the designer's understanding of users' needs and for exploring new design ideas [13]. It is important though, that the sites are representative of the full range of intended users and tasks in order to meet everybody's needs [14].

The designer should get to know the users and question them about their experiences with the product, eliciting what the users like and dislike about the product, what they would like to see changed, and what they think those changes should be. The field study should help the designer to get knowledge about how the finished product will be used, what problems the users are likely to encounter and what is happening in the environment in which the product is going to be used [14].

Techniques used are, for example, interviews, observations and taking notes [4]. A problem with field studies though, is that they can not give a true picture of how the system will perform in reality because end-users are likely to be influenced by the presence of the observer, especially if recording equipment is used. Only if the developers have built up a good relationship with the end-users, they may well be ignored and the field studies are likely to be more useful [13].

Failing to do field analysis often happens when the designers also are going to be users of the application they are building, since they assume that they are average users and think they understand all possible problems from their own experiences. They do not [14].

Usability requirements

The usability requirements must be set to be able to meet the usability goals, discussed further in Section 2.4.1, and associated measures for a particular product. For example, the system needs to be simple so that new users can use the system immediately, and it must be memorable for more frequent users. It also needs to be efficient, to be able to deal with user errors easily and outputs from the system must be clear and unambiguous [4].

Data gathering

An important part of determining the requirements is gathering data. This is also important for the evaluation of the developed system, which will be discussed in Section 2.4.4. The purpose of data gathering is to collect relevant and appropriate data so that the right requirements can be produced. Even if initial requirements exist, these will have to be clarified and expanded upon with help of the data gathered. The designer needs to find out who the users are, what they want out of the system and what kinds of needs they have [4].

Preece et al. [4] presents a number of basic techniques for data gathering. These techniques are questionnaires, interviews, focus groups and workshops[9], naturalistic observation[10], and studying documentation such as manuals. These techniques can all be combined and extended in many different ways. Interviews and questionnaires will be discussed further below.

Which technique to use depends on where in the cycle of iterations the designer is and the kind of information that is needed. Using different techniques is one way of making sure that different perspectives and corroboration of findings are taken into account [4].

Table 1 below gives an overview over these techniques; when to use them, for what kind of data to use them and it shows the advantages and disadvantages with the different techniques.

[9] A group of stakeholders that discuss the issues and requirements [4].
[10] Observing the users in their natural environment [4].

Interviews

There are four main types of interviews; structured, unstructured, semi-structured, and group interviews and the interview questions can be open or closed. Closed questions require the interviewee to select from a limited range of options while open questions accept a free-range response [4].

Structured interviews

In a structured interview, all the questions are predetermined. This type of interview is useful when the goals are clearly understood and specific questions can be identified. The questions should be short and clearly worded and the responses may involve selecting from a set of options that are read out loud or presented on paper. The questions should be refined by asking another evaluator to review them before they are used on the intended users. Typically the questions are closed, which means that they require a precise answer. The same questions should be used with each participant so the study is standardized [4].

Unstructured interviews

In this technique, the interviewer asks the users a series of open-ended questions. The idea is that users will steer the interview in the direction of issues they perceive as important [13]. A benefit with this type of interviewing technique is that it can generate a lot of valuable data since interviewees often mention things that the interviewer may not have considered [4]. This type of interviewing technique is probably most appropriate when the developer has little idea about what the users' concerns actually are or in the early stages when the developer is trying to capture general information about the users, their tasks and their environments [13].

Semi-structured interviews

This technique combines the features of structured and unstructured interviews and use both open and closed questions. The interviewer has a basic script of pre-planned questions, for guidance, so that the same topics are covered with each interviewee. However, the interviewer will probe and follow interesting, relevant directions suggested by the interviewee [4]. With this technique it is possible to gather individual responses from the worker and to gain some indication of their ideas and personal responses [13].

A dilemma with interviews is that what users say are not always what they do. People sometimes give the answers that they think show them in the best light. It may not be possible to avoid this behaviour, but it is important to be aware of it and reduce such biases by using a large number of participants or by using a combination of techniques [4].

Questionnaires

Using questionnaires is a well-established technique for collecting information about current work practices or users' opinions towards a system. They are a good measure of users' attitude toward a system and they can produce huge amounts of useful data, though analysing it can be tedious [13]. One advantage of questionnaires over interviews, is that they can be distributed to a large number of people and hence provide a wide general opinion about a system, however, they are time-consuming to produce and need to be properly tested before they can be given out to the target audience [4].

Table 1. Overview of data-gathering techniques used in the requirements activity, author's interpretation from Preece et al. [4].

Technique	Good for	Kind of data	Advantages	Disadvantages
Questionnaires	Answering specific questions	Quantitative and qualitative data	Can reach many people with low resource	The design is crucial. Response rate may be low. Responses may not be what you want
Interviews	Exploring issues	Some quantitative but mostly qualitative data	Interviewer can guide interviewee if necessary. Encourages contact between developers and users	Time consuming. Artificial environment may intimidate interviewee
Focus groups and workshops	Collecting multiple viewpoints	Some quantitative but mostly qualitative	Highlights areas of consensus and conflict. Encourages contact between developers and users	Possibility of dominant characters
Naturalistic observation	Understanding context of user activity	Qualitative	Observing actual work gives insights that other techniques can't give	Very time consuming. Huge amounts of data
Studying documentation	Learning about procedures, regulations and standards	Quantitative	No time commitment from users required	Day-to-day working will differ from documented procedures

In addition to these techniques, Tognazzini [14] proposes the method brainstorming as a good way for embracing new ideas and throw away old ones. This is a method where a couple of persons sit together and write all their ideas on different pieces of paper. The main purpose of brainstorming is to generate as many ideas as possible so that at least one will lead to a satisfactory solution. The method, function analysis, presented by Landqvist, can then be applied to summarize, refine and to structure the favoured ideas from the brainstorming stage, and to decide where more information needs to be collected [16].

A function analysis is done by dividing the different ideas from the brainstorming session into different groups; one necessary group where all functions that are absolutely decisive for the system to fulfil its central purpose are placed, one desirable group where all functions that are good but not necessary are placed, and one unnecessary group where functions, that does not need to be a part of the system for it to work, are placed [3].

Tognazzini [14] suggests that brainstorming works well in conjunction with scenarios, little "plays" that take place in various parts of the user space explored during the brainstorming.

These will help to consider a wide range of users, in a vide variety of circumstances. With scenarios the developer can create typical situations that users may face, and then build prototypes that will enable exploration of those situations.

Goal Analysis

Goals and tasks have a close relationship since goals are the reason why tasks are performed. Hence, not only should it be the tasks that drive the design, goals should drive it too and the design should help users to achieve their goals with minimum effort. Users often do not care about how they achieve their goals, but only about if the method or task is convenient, reliable and easy to perform. Goals define what the results of the development are going to be, and they can be used as something to measure the progress against [4].

A lot of ideas have to be tried out in order to design something that is original and beautiful. Sometimes this can lead to loosing sight of the problem that has been set out to solve in the first place. The only thing that keeps the designer on track is a clearly expressed set of goals. These goals are also used to evaluate every design decision by constantly comparing if the solution is moving the design closer to or farther away from the goals [17].

Part of the process of understanding users' needs, with the intention to design an interactive system to support them, is to be clear about the primary objective i.e. the usability goals and user experience goals. As mentioned above, these should be identified, clearly documented, and agreed upon at the beginning of the project, as these will form the basis of the product's requirements. These goals will also help designers to choose between different alternative designs and to check on progress as the product is developed [4].

Usability goals are concerned with meeting specific usability criteria, for example, efficiency, and user experience goals are concerned with the quality of the user experience, for example, if the product is aesthetically pleasing [4].

Both usability goals and user experience goals will be described in detail below, but first usability will have to be defined.

Usability

Usability is an essential concept to strive for in interaction design and HCI and it was born out of the desire and need to make things easier and more efficient for the intended users [3]. As mentioned a lot of times in this thesis, improved usability can be sought through greater attention to users and their involvement in the design process [1].

There are numerous of different definitions of usability. According to International Organization for Standardization[11] usability can be defined as *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*.

The simplest definition, though, is the one by Eason [10], *"a system is usable if it is in fact used in practice"*. But, as Löwgren [10], points out, there is a slight drawback with this definition which is that the usability of a system can not be measured until it is installed. This

[11] http://www.usabilitynet.org/tools/r_international.htm#9241-11

becomes particularly serious when the designer wants to develop a system according to the interaction design model described above. Iterating towards a set of usability goals is very hard if consistent measures of how close the design is to these goals can not be done.

Designing usability is an ongoing part of the design process, see Figure 4. An interface is intuitive only if it behaves the way users expect it to, and it can do that only if the designer has been capable of anticipating what assumptions users would make about the product's behaviour [17].

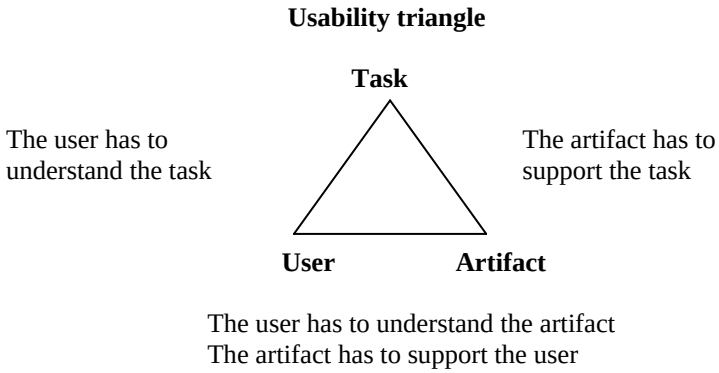


Figure 4. The usability triangle,[12] author's interpretation.

Usability goals

As mentioned above, usability goals are established and agreed upon early in the development process and are then revisited and used to determine the progress as development proceeds. They are generally concerned with ensuring that the interactive product will meet specific usability criteria such as; it has to be easy to learn, effective, and enjoyable to use from users' perspective. Hence, it involves optimising the interactions people have with interactive products to enable them to carry out their activities at work, school, and in their everyday life [4]. The point of usability goals is to formulate measurable criteria on the user aspects of the system [10].

Löwgren [10] has developed a model that can be used to set up usability gaols. He claims that usability is a result of relevance, efficiency, attitude and learnability (REAL).

- *Relevance* - how well the system serves users' needs.
- *Efficiency* - how efficiently users can carry out their tasks using the system.
- *Attitude* - users' subjective feelings towards the system.
- *Learnability* - how easy the system is to learn for initial use and how well users will remember the skills over time [10]. For the memorability part of this, the "ten-minute rule" by can be applied. It proposes that a system fails if a novice user is not able to learn how to use it in under ten minutes [4].

[12] http://www.eat.lth.se/Kurs/Material/MAM061

Preece et al. [4] have extended this list a bit by adding:

- *Effectiveness* - how good the system supports its tasks.
- *Safety* – how well the system is protecting the user from dangerous and undesirable situations.
- *Utility* - does the system provide the right kind of functionality?

User experience goals

As mentioned above, the user experience goals are concerned with the quality of the user experience, what the interaction with the system feels like to the users. It is concerned with ensuring that the interactive product is aesthetically pleasing, enjoyable, fun, entertaining, satisfying, motivating, helpful and so on [4].

However, it is important to be able to make tradeoffs between the usability and user experience goals depending on the context, the task and who the users are. All goals are not going to be able to be met. Getting this balance right requires experience, but also development and evaluation of alternative solutions [4].

2.4.2 Develop alternative designs

Once a set of requirements has been established the design activity begins. This activity is about suggesting ideas for designs that will meet the set of requirements but also about balancing conflicting requirements. As Marc Rettig [18] suggested *"to get a good idea, get lots of ideas"*. The design emerges iteratively, through repeated design-evaluation-redesign cycles involving users. This activity constitutes of two sub-activities; conceptual design, described below, and physical design, described in Section 2.4.2. [4].

Conceptual design

Conceptual design concerns what the system must do in order to achieve its purpose and the necessary structure that is required [7]. It transforms the user requirements and needs into a conceptual model which describes what the product should do, behave and look like, see Section 2.4.2 for more details. It is in this phase the first concrete design activities take place. The conceptual design should be viewed at as a middle step between analysis and prototyping, looking at the overall design of an application or function [4].

During conceptual design, the information requirements need to be considered and the designer needs to ensure that the model caters for the necessary data and that information is available as required for the users to carry out their tasks. Detailed issues of structure and display will more likely be dealt with in the later, physical design activity, but implications arising from the type of data to be displayed may impact conceptual design issues [4].

Repeatedly thinking about different perspectives and considering alternatives can help to expand the solution space and can help to prompt insights. One technique to help generate design ideas is, the mentioned, brainstorming technique and to visualize the design prototyping explained further in Section 2.4.3, storyboards[13] and scenarios are three techniques that can help in exploring ideas, make design decisions and to see if the necessary

[13] A combination of scenarios and user interface sketches [7].

data to perform a task is available. These techniques can also be used to investigate potential conceptual models [4].

An important determinant of the success of a design is the procedural knowledge possessed by users, i.e. their "how-to-do-it" knowledge [7]. Norman [9] states that people do not always behave in a logical way. He suggests that a large number of the difficulties that people experience when performing everyday tasks are caused by a poor relationship between the intention of the actors and the physical actions that can be performed on the object they are working with, and the state of that object. He calls these difficulties the gulfs of execution and gulfs of evaluation.

The gulfs of execution and the gulfs of evaluation

The gulf of execution is the difference between what the user want to do and what can actually be done using the controls that are available. The gulf of evaluation is the mismatch between the expectations and intentions of the user and the real state of the system. The gulf of evaluation reflects the amount of effort that users needs to bring to the system in order to work out what the physical state of the system actually is, and to determine whether or not their intentions and desires have actually been met by the system [6].

Norman [6] argues that the gulfs reflect the distance between how the users believe the system works and the actual physical state of the system and the components that can be adjusted. The gulf is small when the system provides information about its state in a form that matches the way the users think of the system.

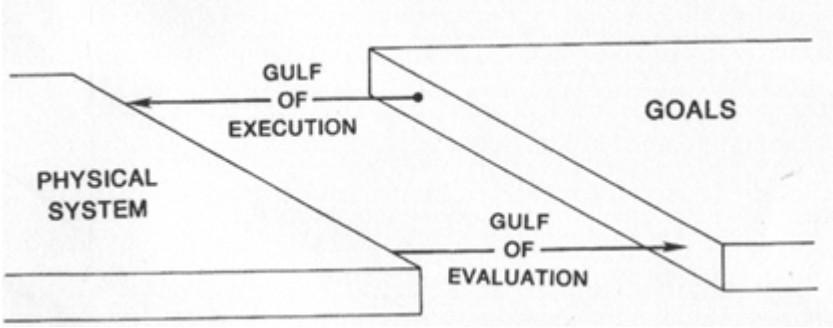


Figure 5. The gulfs of execution and evaluation [6].

It is the designer's obligation to make sure that the gulfs of execution and evaluation is bridged by making things visible on both sides so users know what is possible, how actions should be done and to be able to tell the effects of their actions [6].

Conceptual Model

Preece et al. [4] define a conceptual model as *"a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave, and look like, that will be understandable by the users in the manner intended"*. The basis for designing this model is the set of user tasks the product will support.

The idea of a conceptual model is that users should be able to understand how to operate the system on a general level. If the conceptual model is clear and consistently used in the design, it is easier for the users to apply what they have learnt in one part of the system, in other parts of the system as well [10]. It is the designer's responsibility to develop a conceptual model that is appropriate and understandable to the intended users, and that captures the important parts of the operation of the device. The designer provides a good conceptual model for the users, by using consistency in the presentation of operations and results and with a coherent and consistent system image [9]. Think of a conceptual model of the system as providing scaffoldings upon which to build the bridges across the gulfs, see Figure 6. They allow the user to derive possible courses of action and possible system responses to those actions [6].

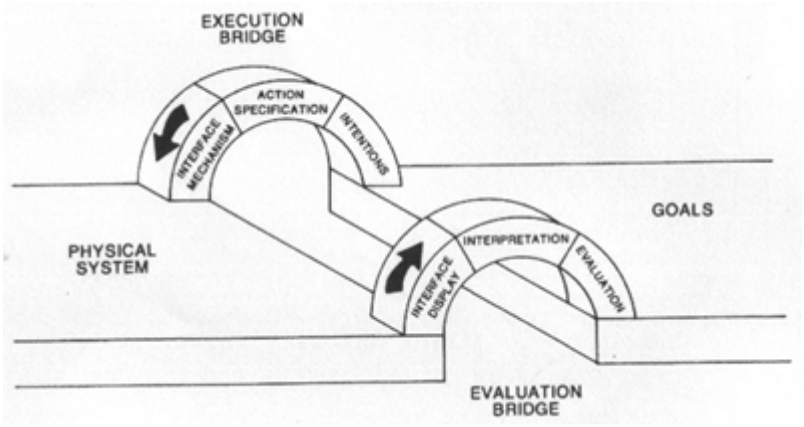


Figure 6. Bridging the gulfs of execution and evaluation [6].

As Preece et al. [4] claims *"the most important thing to design is the user's conceptual model. Everything else should be subordinated to making that model clear, obvious, and substantial. That is almost exactly the opposite of how most software is designed"*.

There is no easy transformation to apply to a set of requirements that will produce "the best" or even a "good enough" conceptual model. Preece et al. [4] states that *"steeping yourself in the data and trying to empathize with the users while considering the issues raised in this section is one of the best ways to proceed. From the requirements and this experience, a picture of what you want the users' experience to be when using the new product will emerge"*.

Norman [9] has developed a framework in order to clarify the relationship between the design of a conceptual model and a user's understanding of it, see Figure 7. Essentially, there are three interacting components; the designer, the user, and the system. Behind each of these are three interlinking conceptual models.

- The design model – the model the designer has of how the system should work.
- The system image – how the system actually works.
- The user's model – the users understanding of how the system works.

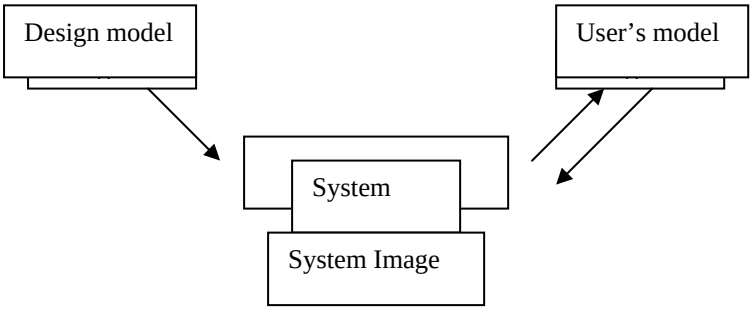


Figure 7. Conceptual model, author's interpretation from Norman [9].

In an ideal world, all these three should map onto each other. Users should be able to carry out their tasks, in the way the designer intended them to, through interacting with the system image, which makes it obvious to do. However, if the system image does not make the design model clear to the users, it is likely that they will get an incorrect understanding of the system, which in turn will make them use the system ineffectively and make errors [9].

Because mental models are inside a person's head, they are not accessible to direct inspection, therefore, in order to ensure that the three components map onto each other, doing iterative testing of the product as it develops is vital [4].

Physical design

Physical design considers more concrete, detailed issues of the design of the interface, compared to conceptual design, such as screen or keypad design, which icons or colours to use, how to structure menus, etc. However, there is no rigid border between conceptual design and physical design. Since interaction design is an iterative process, it is inevitable that some detail issues will come up during conceptual design; similarly, during physical design it will be necessary to revisit decisions made during the conceptual design. One relevant issue though, is that the conceptual design should be allowed to develop freely without being tied to physical constraints too early, as this might inhibit the developer's creativity [4].

There are many aspects to the physical design of interactive products, and all of them can not bee covered in this thesis. Instead, some guidelines and principles of good design in the context of some common interface elements are going to be discussed in the following sections.

Guidelines for physical design

It is important to design the physical interface of the interactive product in a way that will not conflict with the users' cognitive processes involved in achieving a task. To help designers create better designs and to ensure that their products are going to be usable, a wide range of guidelines, principles, and rules have been developed. All of these focus on making the communication between the user and the product as clear as possible and a lot of them are embodied in style guides[14] and standards[15] [4].

[14] A collection of specific design rules and principles from which the rules are derived and are used to ensure a consistent look and feel across a set of applications [4].

Some guidelines are very detailed and are called design rules, while others such as design principles are more abstract and will require interpretation before they are being applied. Design principles will be discussed further below. Yet another form of guidance that will be described in Section 2.4.2 is usability principles. These help to improve the systems usability and help to find usability problems [13].

Design principles

Design principles are one way in which the cognitive models and processes can be put into practical use in the design. They are derived from a mix of theory-based knowledge, experience, and common sense. And the intention with them is to orient designers towards thinking about different aspects of their designs in order to improve them, suggesting what to provide and what to avoid on the interface [4].

However, these principles are not intended to specify how to design an actual interface but to act more like a set of reminders to designers, ensuring that they have provided certain important things on the interface [4].

A number of design principles have been promoted, all focusing on making the communication between the users and the product as clear as possible. Shneiderman's eight golden rules of interface design consist of eight principles and can act as a guideline while creating interface design. However, to be useful they must be interpreted, refined and extended for each environment in which they are going to be used [11].

Eight golden rules of interface design

These principles focus on increasing the user's productivity by providing simplified data-entry procedures, comprehensible displays, and rapid informative feedback that increase feelings of competence, mastery, and control over the system. They can be applied during the design stages or afterwards as an evaluation tool and as a resource of checking for usability [11].

1. Strive for consistency.

Similar situations should be managed in similar ways, i.e. with similar sequences of action. Identical terminology should be used for prompts, menus, information boxes [13]. There should also be a consistency of colour, layout, fonts, and so on throughout the whole application [11].

2. Enable frequent users to use shortcuts.

As the frequency of use increases, so do users' desires to reduce the number of interactions. Things like abbreviations, hidden commands, special keys, and macro facilities are appreciated by users who use the system frequently. Short response times and fast display rates are other things that will be appreciated by these frequent users [11].

3. Offer informative feedback.

There should be system feedback for every user action. For frequent and minor actions, the response can be modest whereas, for infrequent and major actions, the response should be more substantial [11].

[15] A collection of principles and rules to provide designers with a framework based on others' experience [4].

4. Design dialogs to yield closure.

Sequences of actions should be organised to have a beginning, middle and an end. The informative feedback at the end of a group of actions gives users the satisfaction of accomplishment and an indication that it is ready to prepare for the next group of actions [11].

5. Offer error prevention and simple error handling.

The design should be developed in such a way that users cannot make serious errors. However, mistakes are inevitable and the system should be forgiving about the errors made and support users in getting back on track by offering simple, comprehensible instructions for handling errors [11].

6. Permit easy reversal of actions.

Actions should be as reversible as possible. This relieves anxiety and encourages exploration of unfamiliar options since the user knows that errors can be undone [11].

7. Support internal locus of control.

Users feel more comfortable if they feel that they are in control of the interaction with the system, rather than the device being in control. Hence, the users should be the initiators, rather than the responders, of actions. Surprising system actions or inability to produce the actions desired will build anxiety and dissatisfaction among the users [11].

8. Reduce short-term memory load.

The limitation of human information processing in short-term memory[16] requires displays to be kept simple. The application can be made more memorable by making the design model and hence the system image, correspond to the past experiences and expectations of the users. The good relationship between where the control is placed and what it does makes it easy to find the appropriate control for a task, and as a result, there is little to remember. Sufficient training time should be offered to the users so they learn how to control codes, mnemonics and sequences of actions. If needed, provide online- help [11].

"Knowledge in the world and knowledge in the head"

Other principles, such as the ones suggested by Norman [9], are concerned with how to determine what the users should see and do when carrying out their tasks using an interactive product. Norman suggests to *"use both knowledge in the world and knowledge in the head"*. He argues that users will learn better and feel more comfortable when the knowledge required for a task is available externally. But knowledge in the world is only useful if there is a natural, easily interpreted relationship between that knowledge and the information it is intended to express about possible actions and outcomes. Therefore, the design should combine the knowledge in the head with that in the world. He presents the expressions mapping, visibility, affordance, constraints, consistency, and feedback as ways of how this can be achieved.

Mapping

Mapping refers to the relationship between controls and their effects in the world. The mapping is said to be good if the controls of the system and their effects appear natural and

[16] The rule of thumb is that humans can remember 7±2 chunks of information [12].

intuitive to the users. Bad mappings appear if the relations are inconsistent or incompatible [9].

Visibility

The correct things in a system must be visible and they must express the correct message. The more visible functions are the more likely users will be able to know what to do next. Visibility indicates the mapping between intended actions and actual operations. It is the lack of visibility that makes so many computer-controlled devices hard to operate [9].

Affordance

Affordance is defined by Norman [19] as *"a technical term that refers to the properties of objects – what sorts of operations and manipulations can be done to a particular object"* i.e. affordances provide strong clues about how a system is to be operated. When affordances are taken advantage of, the user will know what to do just by looking; no pictures, labels, or instructions are required. However, complex things may require an explanation, but simple things should not. As Norman [9] states "*when simple things need pictures, labels, or instructions, the design has failed*".

Constraints

Constraints limit the number of possibilities of what can be done with the product at a particular moment [7]. This will prevent the users from selecting incorrect options and thereby reduces the chance of making mistakes. Norman [6] classifies constraints into three categories; physical, logical, and cultural. Physical constraints refer to the way physical objects restrict the movement of things by their shape and size, logical constraints rely on people's understanding about actions and their consequences, cultural constraints rely on learned conventions, like the use of red for warning. Once learned and accepted by a cultural group, they become universally accepted conventions.

Consistency

Consistency refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks. A consistent interface is one that follows rules, such as using the same operation for selecting objects which makes the operations easier to learn and use. However, a consistent interface can also mean grouping related functions and placing them under appropriate categories, for example in a menu [14].

Tognazzini [14] states that *"the most important consistency of all is consistency with the user's expectations"*.

Feedback

Feedback is, according to Norman [9], *"sending back to the user information about what action has actually been done, what result has been accomplished"*. Every action should have an immediate and obvious effect otherwise the user may conclude that the action was ineffective and hence repeat it [9]. Feedback can be provided in many different ways. It can, for example, be verbal, visual, aural, tactile or a combination of these [4]. The important thing is that the feedback must be in a form that is easy to understand and that provide information that matches the user's intentions [9].

Usability principles

As mentioned above, usability principles are another form of guidance to help improve a system's usability and for finding usability problems [13]. These are quite similar to the design principles, except that they tend to be more prescriptive. Another difference is the way in which they are used. Design principles tend to be used mainly for informing a design, whereas usability principles are used mostly as a basis for evaluating prototypes and existing systems. Especially, usability principles provide the framework for heuristic evaluation, explained further in Section 2.4.4. When used as part of an evaluation they are called heuristics [4].

Below are the ten main usability principles, developed by Nielsen and his colleagues.[17] Some of them overlap a bit with the design principles described above.

1. *Visibility of system status* - the system should always keep the users informed about what is going on, through providing appropriate feedback within reasonable time.

2. *Match between system and the real world* - the system should speak the users' language, using words, phrases and concepts that are familiar to them, rather than system-oriented terms. The system should also make the information appear in a natural and logical order.

3. *User control and freedom* - the system should provide ways of allowing the users to easily escape from places they unexpectedly find themselves, by using clearly marked emergency exits.

4. *Consistency and standards*— the system should avoid making the users wonder whether different words, situations, or actions mean the same thing.

5. *Error prevention*— the system should, wherever possible, prevents errors from occurring.

6. *Recognition rather than recall* - the system should make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.

7. *Flexibility and efficiency of use* - the system should provide accelerators that are invisible to novice users, but allow more experienced users to carry out their tasks more quickly. This way the system can cater both inexperienced and experienced users.

8. *Aesthetic and minimalist design* - the system should avoid using information that is irrelevant or rarely needed.

9. *Help users recognize, diagnose, and recover from errors* - the error messages should use plain language to describe the nature of the problem and suggest a way of solving it.

[17] http://www.useit.com/papers/heuristic/heuristic_list.html

10. *Help and documentation* - the system should provide information that can be easily searched and provide help in a set of concrete steps that are easy for the users to follow.

Usability trade-offs

A problem that can arise is that trade-offs might have to be made if more than one of the design principles or usability principles are applied in interaction design. For example, the more the interface is constrained, the less visible the information becomes [4].

Designing the interface

This section is going to give some examples of how the design and usability principles described above can be used in the context of interaction design and especially be applied to different interaction styles and different screen elements. It is also going to investigate the importance of understanding the intended users, especially their cognitive aspects, when designing interactive products. But first the two concepts cognition and human cognition are going to be explained.

Cognition

Cognition is what goes on in people's heads when they carry out their everyday activities. It involves cognitive processes, like seeing, reading, thinking, learning, remembering, daydreaming, decision making, writing and talking [4]. Cognition has been described in terms of specific kinds of processes which include:

- *Attention* - involves people's auditory and/or visual senses and is the process of selecting things to concentrate on, at a point in time, from the range of possibilities available. Attention allows users to focus on information that is relevant to what they are doing. The way information is displayed can greatly influence how easy or difficult it is to attend to the appropriate pieces of information. If the information is ordered into meaningful categories by using, for example, colour and by using blank spacing, it is easier to select the necessary information. However, colour should be used sparingly. Too many colours on an interface can result in that it becomes distracting and annoying rather than helping the user attend to relevant information. Interfaces that are plain are much easier to use [4].

- *Perception and recognition* - Perception refers to how information is obtained from the environment, via the different sense organs, like eyes, ears, fingers, and then transformed into experiences of objects, sounds, events, and tastes. It is a complex process that involves other cognitive processes such as memory, attention, and language [4].

  A general design principle is that information needs to be represented in an appropriate form to facilitate the perception and recognition of its underlying meaning. It is important to present information in a way that can be readily perceived in the manner intended. For example, icons should be designed so that they are easy to distinguish from one another and to make it simple to recognize what they are intended to represent [4].

- *Memory* - involves recalling various kinds of knowledge that allow the users to act appropriately. A well-known memory phenomenon is that people are much better at recognizing things than recalling them. Hence, interfaces should be designed in a way that promote recognition rather than recall by using menus, icons, and consistently placed objects [4].

- *Learning* - should be supported in the design interface by encouraging exploration and importantly allowing users to undo their actions. The interface should also constrain and guide users to select appropriate actions [4].

- *Reading, speaking, and listening* - The interface should provide opportunities for making text larger on a screen, without affecting the format, for users who find it hard to read small text. In addition, the length of speech-based menus and instructions should be kept to a minimum. Research has shown that people find it hard to follow spoken menus with more than three or four options [4].

- *Problem solving, planning, reasoning, decision making* - These cognitive processes include thinking about what to do, what the options are, and what the consequences might be of carrying out a given action. The extent to which users engage in the various forms of reflective cognition depends on their level of experience with a domain, application, or skill. Novice users tend to have limited knowledge and will often make assumptions about what to do using other knowledge about similar situations. In contrast, expert users have much more knowledge and experience and are able to select optimal strategies for carrying out their tasks [4].

Several of these processes mentioned above may be involved for a given activity and it is rare for one to occur in isolation [4].

Human Cognition

The way in which an interface is designed can greatly affect how well the users can perceive, attend, learn, and remember how to carry out their tasks [4].

Werner Schneider [20] claims that the human cognition can be divided into two levels; the conscious level and the automatic level, see Figure 8.

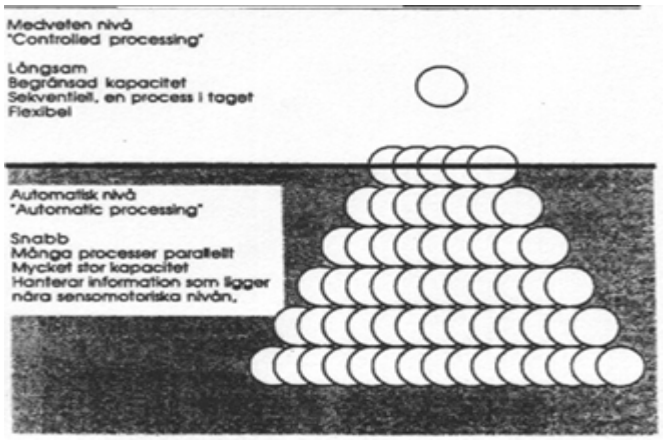


Figure 8. Model of the human cognition [20].

The lower level, the automatic level, has very large capacity. It is specialised to i.e. recognise patterns in colour and shape and to handle automatic movements. At this level, a huge amount of information processes are handled at the same time and millions of mind perceptions are received and managed every second. The upper level, the conscious level, handles one information process at a time. It is on this level that humans can draw conclusions and appraisals of a situation etc. The analytical and the logical ability are large while both the memory and the processor capacity are strongly limited [20].

In all human activity, information processes are conducted in both these levels, at the same time. Thus, the information that is handled consciously is just a fraction of the huge amount of information the human is handling every second. Hence, the way in which the information is presented decides which of the two levels that is going to be involved the most during the read off. Considering that the conscious level is strongly limited in capacity, the information should be presented in a way so that the information, as much as possible, can be handled automatic [20].

A goal with the design of an information system should be to let the users use their creative and problem solving skills to solve, what is the important part of a task instead of figuring out how to interact with the system, i.e. make the interface as obvious as possible to the users. This can be done by displaying the right amount of information at the same time and in a good way on the screen, finish the design, display the overall picture and the details at the same time, allow the possibility of changing between different tasks, display the information in a clear and consistent way, use icons in a correct way etc. [20].

The main benefits of cognitive theories are that they can explain the user interaction and predict the user performance [4].

Interaction styles

There are a lot of different interaction styles that can be used on the interface, such as menus, forms, command languages, direct manipulation, dialog boxes[18] etc. Each has to be designed carefully and with the users in mind. Using several different interaction styles in one interface may be appropriate when the required tasks and users are diverse. Commands may lead the user to a form fill-in where data entry is required or menus may be used to control a direct manipulation environment [12]. Table 2 below shows some advantages and disadvantages with some of these interaction styles.

There are a lot of different style guides that will help to make these interaction styles as clear as possible to the users. Below, some of these style guides have been applied to some of the interaction styles mentioned above.

Menu design

Menus provide users with a choice. It can be a choice of commands or a choice of options related to a command. Menu selection is attractive because it can eliminate training and memorization. Users can easily select an item from the menu with, for example, a pointing device hence reducing the possibility of keying errors [12].

[18] A combination of menus and forms [12].

Menus may be designed as drop-down, linear, pop-up or single-dialog menus. In order to design a menu that is easy to use and provide user satisfaction, some important points must be taken into account such as semantic organization, menu-system structure, number and sequence of menu items, prompting format, graphic layout and design, phrasing of menu items, display rates, and selection mechanisms etc. The primary goal is to create a menu that is comprehensible, sensible, memorable and to offer convenient semantic organization relevant to the users' tasks. Users should have a clear idea of what will happen when they make a selection. One important thing to be taken into account is that opposite operations such as "quit" and "save" should be clearly separated to avoid accidentally losing work instead of saving it [12].

Multiple menus and linear sequences provide simple and effective means for guiding the user through and structuring a decision-making process. The user should be given a clear sense of progress or position within the menu, and should be able to go backward to earlier choices. Care must be taken to match user expectations when choosing the layout of a menu and the grouping and the order of the menu items. One strategy is to place the most relevant option first, or in the upper left in a multiple menu [12].

There is no perfect menu structure that will match everybody's needs. A huge amount of work has been done on exploring the optimal number of items in a menu design, and most studies conclude that breadth is preferable to depth in organizing the menu content [4]. By this it is meant that having a large number of top level menu items with few levels is preferred over having a small number of top level items with many levels. In fact, the results shows that menu trees should be limited to three levels since there appears to be a greater chance of users becoming lost or disoriented when the depth goes to four or five levels [12].

The greatest benefit with menus are that there is a clear structure to decision making, since only a few choices are presented at a time [12]. They will structure and simplify the user's interaction with the system and will reduce the cognitive load, but they can easily be a bit slow [11].

Computerised forms

When data entry is required, menu selection usually becomes cumbersome, and instead form fill-in is appropriate. Forms provide the users with a display of related fields among which the users can move a cursor and enter data where desired. It is important that the field labels are understandable to the users so they will know what to enter in which fields. It is also important to have a logical grouping and sequencing of the fields, consistent terminology and abbreviations, convenient cursor movement, visible space and boundaries for data-entry fields, error messages for unacceptable values, explanatory messages for fields etc. [12].

It should also be clear to the users what they must do when they are finished filling in the fields. Generally, automatic completion when the last field is filled, should be avoided because the users may wish to review or alter field entries. When the user has to enter times or dates there can be confusion about which format to write it in. The best solution is to show an example of the correct way of entry [12].

As with menus, forms will structure and simplify the user's interaction with the system and will reduce the cognitive load, but they too can be a bit slow [11].

Direct manipulation

Direct manipulation means that the complex command language is replaced by direct manipulation of the objects of interest i.e. users can carry out tasks rapidly just by pointing at visual representations of objects and actions, and observe the results immediately. With a careful design, direct manipulation can be appealing both to novices and to frequent users [1].

The main advantage of direct manipulation is that the computer system attempts to model everyday operations more directly than older styles or interfaces, thus making them easier to learn and use hence reduces the cognitive load. A disadvantage though is that fast system responses are needed to support the pointing and direct manipulation processes [1].

Command language

For frequent users, command languages provide a strong feeling of control. Once the users have learnt the syntax they can often express complex possibilities rapidly. However, error rates are high, training is necessary, and retention may be poor. Error messages are hard to provide because of the diversity of possibilities and the complexity of mapping from tasks to computer concepts and syntax [12].

Table 2. Advantages and disadvantages of the five primary interaction styles, author's interpretation from Shneiderman [12].

Interaction Style	**Advantages**	**Disadvantages**
Menu selection	shortens learning reduces keystrokes structures decision making permits use of dialog-management tools allows easy support of error handling	imposes danger of many menus may slow frequent users consumes screen space requires rapid display rate
Form fill-in	simplifies data entry requires modest training makes assistance convenient permits use of form-management tools	consumes screen space
Command language	is flexible appeals to "power" users supports user initiative is convenient for creating user-defined macros	has poor error handling requires substantial training and memorization
direct manipulation	presents task concepts visually is easy to learn is easy to retain allows errors to be avoided encourages exploration permits high subjective satisfaction	may be hard to program may require graphics display and pointing devices

Graphical design

As argued before, interaction design should not just be about usability *per se*, but should also include aesthetic design, such as how pleasurable an interface is to look at. The key is to get the right balance between the usability of the interface and other design concerns, like aesthetics. Badly designed interfaces often make people frustrated and angry [4].

One important thing is to design the interface in a way that matches the users' expectations and task experiences as much as possible rather than force them to understand new principles, tasks, and techniques. This design approach will make the user interface more intuitive and friendly for the users to use [2].

Above in this section some of the visual communication principles were mentioned, for example, designing the screen so that the user's attention is drawn immediately to the relevant points, and using colour, boxing, grouping and motion to aid understanding and clarity. Each screen should be designed so that when the users first see it, their attention is focused on something that is appropriate and useful to the task at hand [4].

The style of an interface, in terms of the colours, shapes, fonts, and graphical elements that are used and the way they are combined, will also influence how pleasurable it is to interact with. This can also have a positive effect on users' perception of the system's usability [4].

Another thing that is important to think about when designing the interface is to make it obvious to the user what is clickable on the screen and what is not. Objects that look like buttons should act like buttons. If images have hot areas, make sure they are distinct from the rest of the image [17].

Properly designed, the graphical design of the interface will increase the visibility, give an obvious affordance, support a good mapping, be used to give good feedback i.e. support the conceptual model for the interface. It will also prevent problems in the interaction, support a good handling of errors, and it can make the interface more visually appealing [21].

Layout

Like all aspects of interface design, screen layout has a functional side, as well as a visual one. How the objects are arranged on the screen determines not only how good they look but how easy they are to understand and use [17]. Good organization helps users to make sense of an interaction and to interpret it within their own context. This can be done by, for example, grouping similar things together or providing separation between dissimilar or unrelated items and by putting frequently used buttons in places that are easy to reach in relation to other items on the screen. Grouping can be achieved in different ways, for example, by placing things close together or by using colours, boxes, or frames to segregate items, or by using shapes to indicate relationships among elements. However, trade-offs may have to be made between sparsely populated screens with a lot of open space, and overcrowded screens with to many and too complicated set of icons etc. If the screen is overcrowded, the users will become confused and distracted. However, too much open space and consequently many screens, can lead to frequent screen changes, and a disjointed series of interactions. One very general but very important guideline, especially for website design is to, *"keep it simple"* [4].

Objects on the screen may serve a variety of different purposes besides being part of a visual arrangement. For example, some are structural, such as windows and borders that delineate

regions for content, some are informational, such as the words and pictures that deliver the content, and some are functional, such as the buttons and other controls for interaction. In fact, an interface should not contain any elements at all, whose purpose is strictly visual [17].

When designing an interface layout it is important that conventions are taken into account. For example, users of western languages are conditioned to scan the screen from top left to bottom right, assume that larger items are more significant, assume that items above have primacy over items below, and look for “more to come” signals at the bottom centre or right. This might not be the way people from other parts of the world do it [17].

Structural sides to an interface

Grids

A grid is a system of two-dimensional guidelines for positioning elements in a layout. It can be a valuable layout guide and helps to ensure accurate alignment of elements within a single screen, and consistent placement of elements that appear on multiple screens [17]. The grid should be invisible [21].

Backgrounds

Although every element on the screen contributes to the overall look and feel of the interface, the background carries the greatest load simply because it fills so much of the screen. The background serves two main purposes; it influences the look, balance, and location of all the elements on the screen and, it fills the empty space so that the other elements are not just floating objects. In many cases a purely decorative or plain background is all that is needed [17].

Windows and panels

A window or a panel can be any distinct region of the screen. Windows usually hold media, while panels may enhance the structural features of the design, or add depth or colour. Frames and borders around windows should only be used when they serve an integral purpose in the design. In most cases, text, images etc. work best when integrated with their surrounding area, free of unnecessary boxes or frames. Windows and structural panels delineate a region of the screen for a particular purpose or type of content and they anchor design elements within the structure so these do not appear to be floating [17].

Functional sides to an interface

Interface elements

A good interface is built up by a set of elements that should work together to produce a coherent interface with a sense of continuity and consistency. This feel of consistency can be created with colour, position etc. [17].

Buttons and other controls

Buttons and other controls, manage the objects users interact with on the screen, hence they need to be clear and unambiguous. A control can be any part of a screen or region of an image and has to reveal its purpose at first glance. It should also be proportional in importance to the function it represents [17].

Icons

An icon is a special type of button that symbolically depicts what it does [17]. A well designed icon is one that is designed so it is immediately recognizable to the users and so that they are distinguishable from one another. To create a well designed icon takes time! At a simple level, icons should always be designed with existing traditions or standards in mind, and certainly not contradict these [4].

If an icon has a label beneath it, it gets bigger and bigger targets can be accessed faster and the users are less likely to select the wrong one. Furthermore, icons that do not have labels are likely to be placed closer together so they are more crowded [4].

Icons are best used for concrete concepts and are most effective as a miniature representation of the physical object to which they refer. Advantages with icons are that they are visually more distinctive than a set of words and they can represent a lot of information in a small space. One disadvantage of icons though, is that, if designed poorly, they require the user to learn and remember their meaning [1].

It has been found that an interface based on menus and icons is preferred by most people over a strictly alphanumeric interface because, when these graphic features are properly designed, they seem more natural, are easier to learn and use, require little memorizing, hence result in fewer mistakes [1].

Informational side to an interface

Text

The design and the layout of the text are essential since it usually constitutes one of the main parts of an interface. Even when words are used decoratively as part of a background or an image, their meaning matters. Text on the screen has to be easy to read and it has to work in the interface. The appropriate typeface, size, spacing, colour, and format must be chosen carefully [17]. Some ground rules are to use a maximum of two typefaces in one interface, use white space, to use sanserif, and to use the same way to emphasize [21].

Each screen or window should only contain the information that is really needed for the users to perform the expected tasks at that point in the interaction. The temptation to provide additional data just because it is available should be avoided, since extra clutter clearly degrades users' ability to extract the relevant information [2].

Images

An interface might include images of all types like scanned photos, cartoons, computer-rendered three-dimensional objects, and so on. There are no right or wrong types to use. What matters is that the integration of the images should support the design [17].

Visual side to an interface

Colour

Colour can be pleasing, motivating, gain attention and the effectiveness of the graphical interface will be greatly increased if colours are applied with its limitations in mind. One of these limitations is that different colour combinations can make the information hard to read, for example, blue and red, blue and black, yellow and white. Another limitation is that part of

the user population is colour-blind. Yet another is that too many colours can make the display extremely confusing [1].

Colour can be used to help in formatting, i.e. similar colours can be used to group related items. Some guidelines to follow are to use colour conservatively, use the same colour-coding rules throughout the system, and to, where possible, use common denotations of colour for example, red for danger or stop, green for ok or go etc. [2].

Table 3. Colours' different effects on humans,[19] author's interpretation.

	Spatial effect	*Termic effect*	*Psychological effect*
Green:	distant	cold	restful
Yellow:	close	warm	stimulating
Blue:	distant	cold	restful
Red:	very close	warm	very stimulating/not restful
Brown:	claustrophobic	neutral	stimulating

Designing for the web

When designing for the web, the kind of good interaction design described in this section need to be exhibited, but in addition some specific requirements are needed. Nielsen has suggested a set of evaluation criteria specifically for the web which are described further in Section 2.4.4. [4].

The key design issues that are different for websites than from other interaction designs are captured very well by three questions; where am I? What's here? Where can I go? Every web page should be design with these three questions in mind and the answers to them must be clear to the users [22].

Jeffrey Veen [22] expands these questions a bit. He suggests that a simple way to view a web page is to divide it into three areas, see Figure 9.

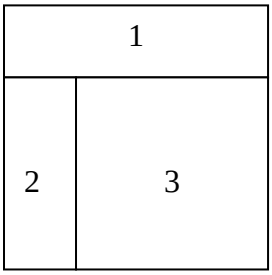


Figure 9. The three main areas every web page should have, author's interpretation from Veen [22].

Across the top area, 1, would contain the answer to "where am I?". Navigations or menus are placed in the area down the left-hand side, 2, to allow users immediately to see what else is

[19] http://www.eat.lth.se/Kurs/Material/MAM061

available on the site. This are answers the question “where can I go?”. The third area is the content area, 3, which contains the most important information, and answers the question “what’s here?” [22].

The content for web pages must be designed differently from standard documents, since the way users read web pages is different. On web pages, the content should be short and precise using headlines to capture the main points of a paragraph. This way the users can scan the page for relevant information, instead of having to read it in detail. It is important, as mentioned above, to keep the screens uncluttered so that the users can find their way around and clearly see what is available [4].

Use graphics sparingly since download times are critical for the success of a website. If the users have to wait too long for a page to show, they will move on to somewhere else. The key is getting the right balance between aesthetic appeal, and the right kind and amount of information per page [4].

Nielsen suggests having few graphics on the welcoming page and then, only when the users explicitly ask for seeing pictures of products or maps, will these be displayed. It is quite common to use thumbnails[20] as links [4].

If menus are used, the most commonly used functions should be at the top, to avoid frequent long scans and scrolls and the names need to be short, clear, and unambiguous [4].

Choosing among alternative designs is about making design decisions. The basis for choosing between the different design options can be set by letting users interact with the different design options and by discussing their experiences, preferences and suggestions for improvement. Letting the users interact with the product like this is fundamental to a user-centered approach to development. This in turn means that the designs must be available in a form that can be evaluated by the users, prototypes have to be built. This is what the next section will be about [4].

2.4.3 Building interactive versions of the designs

The design of a system requires continuous evaluation. So, after multiple design alternatives have been raised, interactive version of the design has to be built for representative users to interact with, conducting real tasks in a realistic context of use, in order to be able to evaluate if the design is right. This does not mean that a software version is required, paper-based prototypes will work just as well [12]. As Löwgren [10] states *“an idea is hard to evaluate; a prototype can be evaluated in several different ways”*.

Prototyping

A prototype attempts to approximate the finished product in a way that is as realistic as possible, without actually implementing the product [16]. It allows the users to interact with an envisioned product and it is a great way to try out the requirements about what the system should and should not do and to make sure that the user-interface design goals are met. It eliminates the possible uncertainties and misunderstandings regarding the design and allows the designer to gain some experience of using the product in a realistic setting [4].

[20] Miniaturized versions of the full pictures [4].

Prototypes answer questions and help to find possible problems and solutions at an early stage of the design, before the specification and implementation phases have started. Prototypes are helpful when testing out the technical feasibility of an idea, to clarify some vague requirements and to check that a certain design direction is compatible with the rest of the system development. These results can then help designers to choose between alternative designs [4]. If the prototype is tested and found to be unsatisfactory in some way, it has to be redesigned. If time and budget permit, the redesigns are tested again. Allocating plenty of time to product testing and redesign helps ensure a well designed product, i.e. a product that will meet the needs and requirements of the users [16].

Löwgren [10] proposes two different kinds of development philosophies. In the most common approach, called evolutionary prototyping, the prototype is evolved into the final product. In the alternative approach, called revolutionary prototyping, prototypes are used only to develop consensus on the functionality and design of the system. The delivery system is then implemented from scratch, using the prototype only as a specification.

Prototyping allows the users to evaluate the system as it develops and user feedback can then be used in the further development of the system [13]. The goal with prototyping is to get through as many iterations and redesigns as possible during the design phase in order to improve the system [18].

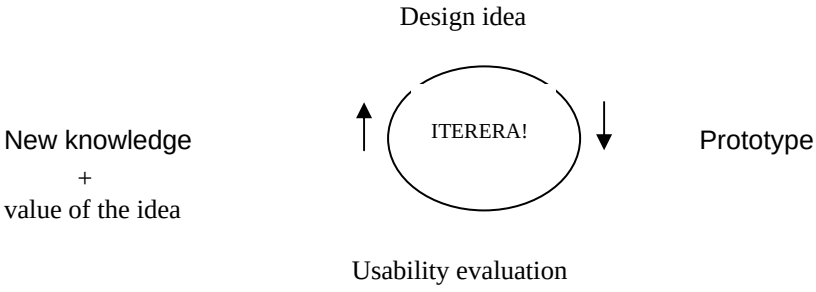


Figure 10. Prototyping circle,[21] author's interpretation.

The more iteration that is done in the prototyping circle, the better the assurance is that the proposed design idea works, and the better the final product will be. However, the design idea needs to be good for this cycle to work, otherwise the designer can get stuck between two bad alternatives.[22]

When the design has been around the iteration cycle enough times for the designer to feel confident that it fits requirements, everything that has been learned through the iterated steps of prototyping and evaluation must be integrated into the production of the final product, see Section 2.4.5. [4].

A prototype can be anything from a paper-based storyboard through to a complex piece of software i.e. low-fidelity respective high-fidelity prototypes. In general, low-fidelity prototypes are used early in design and high-fidelity prototypes are used later on in the design [4].

[21] http://www.eat.lth.se/Kurs/Material/MAM061
[22] http://www.eat.lth.se/Kurs/Material/MAM061

Low-fidelity prototyping

A low-fidelity prototype is one that does not look very much like the final product but still can give a good feeling for the systems interactive behaviour [3]. It is very useful because it tends to be cheap, simple, and quick to produce hence it is cheap, simple, and quick to modify [4].

A lo-fi prototype is often made out of paper or cardboard which is very different from the intended final version, hence it is for exploration only and is never intended to be kept and integrated into the final product [7]. A great thing about a paper prototype is that its hand made appearance forces users to think about content rather than appearance [18].

However, one drawback with this type of prototype is that some things are hard to shape and it is hard to prepare and to keep track of all the components that are needed in order to create a dynamic system [3].

Different types of lo-fi prototyping methods are paper prototypes, storyboarding, sketching, and wizard of Oz[23] [4].

High-fidelity prototyping

High-fidelity prototypes behaves, feels and looks much more like the final thing because they use materials that are expected to be in the final product, hence they yield the most reliable evaluation results [10]. Disadvantages with these types of prototypes, though, are that they often take a long time to build and change. Another drawback is that the product may be perceived as the real thing by the users that it is tested on, hence reviewers and testers might comment on "fit and finish " issues rather than giving feedback on the relevant things like the flow of the conversation, the general layout of the controls, the terminology, etc. Instead the users might criticise the designer's choice of fonts, colour combinations, or button size [18].

Yet another drawback with this prototype technique is that the developers may get very attached to their work because it took so long time and was so hard to implement, hence they might not want to do drastic changes to it even though it might be needed [18].

A hi-fi prototype can, for example, be a computer simulation of the product.

Table 4. Relative effectiveness of low- vs. high-fidelity prototypes, author's interpretation from Preece et al. [4].

Type	Advantages	Disadvantages
Low-fidelity prototype	• Lower development cost. • Evaluate multiple design concepts. • Useful communication device • Address screen layout issues. • Useful for identifying market requirements. • Proof-of-concept.	• Limited error checking. • Poor detailed specification to code to. • Facilitator-driven. • Limited utility after requirements established. • Limited usefulness for usability tests. • Navigational and flow limitations.

[23] A form of prototyping in which the user appears to be interacting with software when, in fact, a member of the development team is responding to the user's actions [7].

High-fidelity prototype	• Complete functionality. • Fully interactive. • User-driven. • Clearly defines navigational scheme. • Use for exploration and test. • Look and feel of final product. • Serves as a living specification. • Marketing and sales tool.	• More expensive to develop. • Time-consuming to create. • Inefficient for proof-of concept designs. • Not effective for requirements gathering.

2.4.4 Evaluating designs

Preece et al. [4] defines evaluation as *"the process of systematically collecting data that informs us about what it is like for a particular user or group of users to use a product for a particular task in a certain type of environment"*.

Evaluating what has been built is the most central thing in interaction design. It facilitates the understanding between designers and users and should be done continuously during the whole design process to be able to ensure that the product matches the set requirements, that usability and user experience goals are met and that the product is usable and acceptable to the intended users. This can be achieved by addressing a user-centered approach to design, which requires a lot of user involvement throughout the whole design process [4].

There are many different ways to apply this user-centered approach to evaluation, for example, through interviewing the users, observing them, testing them using performance tasks, asking them to fill in questionnaires etc. The findings from these different ways of engaging and eliciting knowledge from users are then interpreted with respect to ongoing design activities [4].

Evaluation plays an essential role in every phase of the design, but may differ depending on where in the process it is performed. During the early stages of design, evaluations tend to be done to predict the usability of a product, check the understanding of the users' requirements are right and to test out ideas quickly and informally [7]. It is important, during these first stages of assessment of the system, that as many of the usability faults as possible are caught because the earlier they are caught the easier and cheaper it is to fix them [13]. Later on in the design process, the evaluation shifts to focus on identifying user difficulties, improving an upgrade of the product [7].

Many factors need to be taken into account when planning an evaluation and selecting appropriate methods including the purpose of the evaluation, the involvement of users, the kind of data collected and how it is analysed, the stage of the system development, and the practical constraints associated with actually doing the evaluation [1].

A key factor is the purpose of the evaluation. Four main reasons for doing evaluation can be identified:

- *Understanding the real world* – how well does the design work in the real world?
- *Comparing designs* – which design is the best?
- *Engineering towards a target* – is the design good enough?
- *Checking the conformance of the product to a standard* [7].

To split up the varying evaluation methods, they can be classified into five groups: observation and monitoring, experimenting and benchmarking, collecting users' opinions, interpreting situated events and predicting usability [7]. All these techniques can be used to address different issues at different stages of the design process. They complement each other and together they provide a broad picture of the system's usability, seen from different perspectives. Which technique to choose depends on what stage of the design process the design is in and the particular questions to be answered [4].

Several different kinds of evaluation technique depend on some form of observation and monitoring of the way the users interact with a product or a prototype. It may take place as part of a usability test which is explained further in Section 2.4.4. [7].

When using experiments and benchmarking as part of the evaluation it is usually more rigorously controlled than when just observing and monitoring. This is because the data that is collected will be analysed quantitatively to produce metrics to guide the design. As well as examining users' performance, it is important to find out what users actually think about the system, because no matter how good the users' performance are when using the product – if they do not like it they will not use it. Surveys using questionnaires and interviews are great ways of collecting users' attitudes toward the system [7].

The purpose of interpretative evaluation is to enable designers to better understand how users are going to use the system in their natural environments and how the use of these systems will integrate with other activities. The data is collected with the intention of causing as little disturbance to the users as possible [7].

The aim of predictive evaluation is to predict the kinds of problems that the users will encounter when they are using the system. This is done by experts without actually testing the systems with the users, hence the process is relatively inexpensive and quick. A dilemma with this model though, is that only predictable behaviour can be predicted which makes it difficult to use this model to evaluate how systems will be used in real-world contexts, given that most people are unpredictable in the way they behave. This evaluation method can, however, provide useful estimations for comparing the efficiency of different methods of completing tasks, particularly if the tasks are short and clearly defined [4].

The GOMS model is the most well-known predictive modelling technique in human-computer interaction and it will be described in detail further down. The keystroke level model, and Fitts' Law are two other predictive techniques that can be useful when determining whether a proposed interface, system or keypad layout will be optimal [4]. These will be described further down. Cognitive walkthrough and heuristic evaluation is two other predictive evaluation techniques that are also going to be investigated further down.

Often, more than one of the evaluation methods mentioned above, will be used so that the results can be reviewed together to give a better overall picture of the system's usability [7]. Some advantages and disadvantages can be viewed in Table 5, and different evaluation methods are described below.

Table 5. Some advantages and disadvantages of evaluation techniques, author's interpretation from Preece et al. [7].

Method	Advantages	Disadvantages
Observing and Monitoring	Widely applicable highlights difficulties	Can affect users' behaviour
Experiments and benchmarking	Provides measurements to guide design	Requires expensive facilities
Users' opinions	Inexpensive	May get low response rate
Interpretive	Reveals what really happens in the field	Requires sociological expertise
Predictive	Most forms do not require a working system	Some forms have a narrow focus

The results from the evaluation are then fed back into the design. As you may have noticed, some of the techniques used in evaluation are the same as the ones used in the activity of establishing requirements and identifying users' needs, but they are used in a different way here. The goal of evaluation is to assess how well a design fulfils the users' needs and whether users like it or not [4].

Evaluation Methods

When practicing a user centered design it is important to evaluate the system repeatedly. Numerous of evaluation methods can be used, some of them are presented below.

"Quick and dirty" evaluation

This type of evaluation can be performed at any stage of the design process to get fast feedback from users about, for example, a design idea or to check that the designer's ideas are in line with the users' needs [4].

Usability testing

Usability tests are a central aspect in interaction design. The goal with this evaluation method is to obtain objective performance data that shows how usable a system or product is in terms of usability goals, such as ease of use or learnability [4]. The usability tests help designers to pick up problems that they can not find themselves and help to test predicted problems gathered from, for example, a heuristic evaluation [23]. Heuristic evaluation will be discussed further down. More generally, usability testing relies on a combination of techniques including interviews, observation, questionnaires as well as user testing which is a central component to usability tests. Together these techniques provide a much broader picture of the user's interaction than any single technique would show on its own [4]. Interviews, observations, questionnaires have all been explained before so it is just user testing that is going to be investigated further in this section.

As Helander et al. [2] states *"you won't know whether the design is working right until you start testing it"*. User tests should be performed from the very beginning of the development process, and throughout it, i.e. let the intended users carry out real tasks using early versions of simulations and prototypes of user interfaces.

A basic premise is that, no matter how experienced or smart you are, you cannot get it right the first time and as Helander et al. [2] sates *"plan to throw one away"*.

User testing is held by developers in controlled laboratory-like conditions to test whether the product they are developing is usable and acceptable to the intended user population to achieve their tasks [24]. The developer studies the users' interaction with the system or the prototype as they try to use it to carry out a set of short tasks [3]. In addition the developer measures the time it takes for a typical user to complete clearly defined, typical tasks and record the number and type of errors they make. Sometimes the routes that users take through tasks are also noted, particularly in web-searching tasks [4]. A rule of thumb here is that if two out of three users have the same problem, it needs to be taken care of [3].

To make sense of the data gathered, answers to user-satisfaction questionnaires, interviews and key stroke logs are of great help, which is why these techniques are used along with user testing in usability studies [4].

Bruce Tognazzini[24], comments that *"iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don't have user-testing as an integral part of your design process you are going to throw buckets of money down the drain"*.

A problem with user testing though is that it can be hard and costly to get a hold of users [3]. Another is that the test conditions are rather artificial and not representative of the real world. It is however good for fine-tuning product upgrades [7].

Preparations for a usability test

There are many things to consider before doing a usability test. To get the most out of the test it is very important to be prepared i.e. ensuring that the conditions are the same for each participant, to ensure that all the documents for recording information are ready and have been tested i.e. having questionnaires ready and thought trough different task scenarios [13]. A good test scenario is, according to Dumas and Redish, short, in the user's words and unambiguous. It should give the users enough information to do their task and it should be directly linked to the tasks and concerns [23].

One of the most important preparations is to define goals for the test [23]. These goals will help determine whether the evaluation method will be "quick and dirty ", performed in a controlled environment or in the field and they will form the basis for the evaluation and interpretation of the data. A great help in identifying the necessary steps for a successful study is working through the DECIDE framework [4].

DECIDE

To be able to do an evaluation that is well-planned you have to have clear goals and appropriate questions. The DECIDE framework provides a checklist of the main issues that needs to be considered when planning an evaluation, hence it can be used to guide the evaluation [4].

[24] http://www.asktog.com

1. Determine the overall goals that the evaluation addresses.
2. Explore the specific questions to be answered.
3. Choose the evaluation paradigm and techniques to answer the questions.
4. Identify the practical issues that must be addressed, such as selecting participants.
5. Decide how to deal with the ethical issues.
6. Evaluate, interpret, and present the data.

1. Determine the overall goals that the evaluation addresses.

This is the first step in planning an evaluation, since these goals should guide it and influence which evaluation paradigm to use. These goals could for example be to check that the users' needs have been understood or to determine how usable a product is [4].

2. Explore the specific questions to be answered.

Questions that must be answered to satisfy the goals have to be identified in order to make the goals operational. It can be questions like; is the system difficult to navigate? Is the feedback confusing or maybe insufficient? Is the terminology confusing because it is inconsistent? Is response time too slow? [4]

3. Choose the evaluation paradigm and techniques to answer the questions.

Having identified the goals and main questions, the next step is to choose the evaluation paradigm and techniques. Combinations of techniques can be used to obtain different perspectives. Some general questions about the techniques also need to be asked though, such as; is the technique reliable? Will the approach measure what is intended? [4]

4. Identify the practical issues that must be addressed, such as selecting participants.

There are many practical issues to consider when doing any kind of evaluation such as who the users are going to be, what equipment to use, schedules and budgets. It is important to identify these issues before starting the evaluation [4].

5. Decide how to deal with the ethical issues.

There are a lot of different organizations that provide ethical codes that should be uphold if their activities involve human beings. For example, people's privacy should be protected, which means that their name should not be associated with data collected about them, personal records containing details about education, health, financial status, employment, and where participants live should be confidential. The general rule to remember when doing evaluations is *"do unto others only what you would not mind being done to you"* [4].

6. Evaluate, interpret, and present the data.

Decisions are needed about what data to collect, how to analyse it, and how to present the findings to the development team. The type of data collected is to a great extent determined by the technique used for the evaluation, but there are still some choices. For example, should the data be treated statistically? How should the data be analysed and represented? [4]

When all preparations are done for a usability test, it is always best to do a small study, a pilot study, before attempting the main study [7]. The aim of this pilot study is to help to ensure that the usability test is well designed and that it is likely to be successful, for example, checking the equipment and making sure that the questions in a questionnaire are clear. Pilot tests are important to identify potential problems in advance so that they can be corrected and

for determining whether the experimental design is suitable before time, effort and money are invested in a full-scale evaluation [4].

The Participants

Dumas and Redish [23] states *"to get useful results from a usability test, you must know the users – and potential users – of the product"*.

Hence, the participants in a usability test must represent the targeted user group. This group may be obvious because it may well be the same as the group identified during the process of requirements gathering and specification. However, some designs may require targeting different types of users at different occasions so it is advisable to check that the correct target group has been identified [13]. This is also important to ensure that the findings from the user test can be generalized to the rest of the user population [4].

Deciding how many users to test depends on schedules, budgets, facilities and participants available but also on which technique that is being used. Many professionals recommend that 5-12 testers is enough but there can be fewer for example when doing the "quick and dirty" tests where normally just one or two users are present [24].

The Test

The type of prototype available for testing as well as what kinds of goals and questions that are to be tested will determine what kind of test that needs to be prepared. For example, the breadth and complexity of the tasks to be tested will be influenced by whether it is a paper prototype, a simulation, or a limited part of a system's functionality that is going to be tested. Typically, tests take place in laboratory-like conditions that are controlled and sometimes a video camera is used to help explain why the users did what they did [4].

The goal of a usability test, is not to walk the test subjects through the prototype, but to let them interact with it on there own and observe what they do; what they manage easily, where they get confused, what they try that does not work, where they simply give up etc. The prototype might include a main menu that shows all major topic areas, worst-case screens[25], alternative designs etc. so the designer can ask for comments on possible solutions and which design they prefer. Specific scenarios can be used to test out specific things such as predicted problems and so on [17].

Observation involves watching and listening to users while they interact with the software. This can tell an enormous amount about what the users do, the context in which they do it, how well technology supports them, and what other support that is needed, because when observing, designers will see users do things they never expected them to do [4]. The purpose of observing users is to see what parts of the product that might be difficult or ineffective. Therefore, if a participant is struggling or making mistakes, the difficulties should be blamed on the faulty software design, not on the participant's inexperience or lack of intelligence [14].

It is important to know exactly how the users are experiencing the product that is being tested. Techniques such as interviews and questioners have been mentioned before but another way

[25] Those screen where the users are likely to experience difficulty [17].

to get this information is to ask the user to think out loud during the whole test i.e. to say out loud everything that they are thinking and trying to do. This technique provides valuable insights about how people are operating with the system and their strategies for carrying out the task [13].

As Tognazzini [14] states *"'user observation through thinking out loud' results in our being able to 'see inside' our user's conscious minds, to understand what errors in process are taking place"*.

There may be a problem with this technique though, since it may affect the way in which users operate the system. How well the method will work depends on how the user feels about talking of what is being done. The more comfortable the user feel about this the more likely this method will work [13].

Generally, each task in the user test lasts between five and twenty minutes. Tasks are often straightforward, but occasionally they can be more complex. Easy tasks at the beginning of each testing session will help build users' confidence [4].

It is a good idea to keep a record of what is found out during the test. That way, the designers will have documentation to support their design decisions and they will be able to see trends in users' behaviour [14]. The data that is collected should measure the user performance while doing tasks [4]. To do this, Shneiderman has come up with five measurable human factors which all are central to evaluation.

1. *Time to learn*; how long does it take for a typical user to learn how to use the system to do a given set of tasks? [12]

2. *Speed of performance*; how long does it take a typical user to carry out a given set of tasks using the system? [12]

3. *Rate of errors by the user*; how many errors does a user make while carrying out the benchmark tasks and how serious are they? Error making is such a critical component of system usage that it deserves extensive study, although time to make and correct errors might be incorporated into the speed of performance [12].

4. *Retention over time*; how well do the users maintain their knowledge about how to use the system on a task that they have has not done for some time? Retention may be linked closely to time to learn and how often the product is being used [12].

5. *Subjective satisfaction*; how much did users like using various features in the system? The answer can be found by interviews or by written surveys that includes satisfaction scales and space for comments [12].

Evaluation of the Test

When the usability test has been done all the data gathered from the test should be analysed to see if the usability goals have been met [4]. The results of Shneiderman five measurable human factors are essential here. Although every designer would like to succeed in each of these five categories, forced tradeoffs often have to be made. What tradeoffs to make depend

on what kind of application that is being developed. Requirement documents should make clear which goals that are primary [12].

Designers might be surprised when they see that sometimes areas of an application where problems were expected have none, while areas thought to be perfect are fatally flawed. This is why testing can save time, rather than cost time because the designer does not need to work on things that are not broken [14].

After the results have been examined, the problems found should be fixed and then the product should be tested again to see how the changes affect the users' performance. This iteration should be done until satisfaction [14]. One issue with this though, no system is ever perfect; there are always things, revealed during user testing that can be improved in one way or another. Normally, schedule and budget constrains determine when to stop testing [4].

Heuristic evaluation

Heuristic evaluation is a technique where a small group of usability experts evaluate a design, using a specified set of usability principles known as heuristics [10]. These heuristics closely resemble the high-level design and usability principles and guidelines discussed in Section 2.4.2 e.g., making designs consistent, reducing memory load, and using terms that users understand, but when used in evaluation, they are called heuristics [4].

Using a set of heuristics, expert evaluators work with the product role playing typical users and noting the problems they encounter. These heuristics guide them to focus on key usability issues of concern and help them evaluate whether user-interface elements, such as menus, navigation structure, dialog boxes etc., conform to the principles [4].

As mentioned above, when used in evaluation design and usability are usually referred to as heuristics. This term emphasizes that something has to be done with them when they are applied to a given problem [4].

Below Nielsen's heuristics, explained in Section 2.4.2 have been expanded to include some of the questions addressed when doing evaluation;

1. *Visibility of system status* - Are users being informed about what is going on? Is appropriate feedback about a user's action provided within reasonable time? [4]

2. *Match between system and the real world* - Is the language used at the interface straightforward? Are the words, phrases and concepts that are used, familiar to the user? [4]

3. *User control and freedom* - Does the interface allow users to easily escape from places they unexpectedly find themselves in? [4]

4. *Consistency and standards* - Is there a consistency in the way similar actions are performed? [4]

5. *Error prevention* - Is it easy to make errors? If so why and where? [4]

6. *Recognition rather than recall*- Are objects, options and actions always visible? [4]

7. *Flexibility and efficiency of use* - Have, for example, shortcuts been provided that allow more experienced users to carry out their tasks more quickly? [4]

8. *Aesthetic and minimalist design* - Is any irrelevant or unnecessary information provided? [4]

9. *Help users recognize, diagnose, and recover from errors* - Are the error messages helpful? Do the error messages use a simple language to describe the problem and suggest a way of solving it? [4]

10. *Help and documentation* - Is help information provided that is easy to follow and easy to search? [4]

However, some of these core heuristics are too general hence, different combinations and different types of heuristics are needed to evaluate different types of applications and interactive products. This means that evaluators must develop their own set of heuristics that is more closely tailored to their product. One way of doing this is to tailor the heuristics proposed by Nielsen and by referring to design guidelines, market research, and requirements documents. But producing questions suitable for heuristic evaluation often results in more of them, so trade-offs may have to be made. Exactly which heuristics that are the best and how many that are needed are debatable and depend on the product [4].

When evaluating a commercial website Nielsen[26] suggests that the following heuristics are more useful. Together they form the acronym HOME RUN:

- High-quality content
- Often updated
- Minimal download time
- Ease of use
- Relevant to users' needs
- Unique to the online medium
- Netcentric[27] corporate culture [4]

Because users and special facilities are not needed when doing heuristic evaluation it is, compared to user testing, less expensive, quicker and more flexible. But heuristic evaluation should not be thought of as a replacement for user testing, rather as a complement since these two techniques often reveal different usability problems [4].

Walkthroughs

Walkthroughs are an alternative approach to heuristic evaluation for predicting potential user problems without doing user testing. They involve walking through a task with the system to discover problematic usability features. They are very focused, since they require attention to the smallest details of the user's tasks, and are therefore suitable for evaluating small parts of a system. Most walkthrough techniques, such as cognitive walkthrough, do not involve users. Others, such as pluralistic walkthroughs, involve a team that includes users, developers, and usability specialists. This technique will however not be explored further in this thesis [4].

[26] http://www.useit.com
[27] A company that uses Internet technology and a Web precence in order to further its business initiatives. http://www.netlingo.com/lookup.cfm?term=netcentric.

Cognitive walkthrough

The cognitive walkthrough is based on a theory of human learning and action [10]. It is carried out by an expert that pretends to be the user but at the same time has an experts understanding of the problems the user might encounter at each stage of an interaction and what the difficulties with the system are likely to be [13].

As Nielsen and Mack [25] states *"cognitive walkthroughs involve simulating a user's problem-solving process at each step in the human-computer dialog, checking to see if the user's goals and memory for actions can be assumed to lead to the next correct action"*.

To be able to carry out a cognitive walkthrough a close and considered appreciation of potential user behaviour is required i.e. the expert must know what sort of knowledge the potential users are likely to bring to the system. Without this knowledge it is impossible for the expert to predict where there are likely to be difficulties [13].

The steps involved in cognitive walkthroughs are:

1. The characteristics of typical users are identified and documented, including their level of experience and any assumptions made about them, and sample tasks are developed that focus on the aspects of the design to be evaluated. A description or prototype of the interface that is going to be developed is also produced, along with a clear sequence of the actions that the users need, to be able to complete a given task using the given system [4].

2. A designer and one or more expert evaluators then gather to do the analysis [4].

3. The evaluators walk through the action sequences for each task, placing it within the context of a typical scenario, and as they do this they try to answer questions like:
   - Will the user know what to do to achieve the task?
   - Will the user see how to do it?
   - Will the user understand from feedback whether the action was correct or not? [4]

4. As the walkthrough is being done, a record of critical information is collected in which:
   - The assumptions about what would cause problems for the users and why, are recorded.
   - Notes about design changes and other issues are made.
   - A summary of the results is compiled [4].

5. The design is then modify to fix the problems that are presented [4].

The strengths of this technique are that it focuses on users' problems in detail, yet users do not need to be present, nor is a working prototype necessary which makes it relatively cheap and quick to carry out. However, it is very time-consuming and laborious to do and it does require that the expert understands the tasks that are to be done and be able to break these down into separate stages. The expert must also be able to accurately predict likely user performance and must have an understanding of the user's probable cognitive abilities. Neither of these things is easy. As Faulkner [13] states *"a cognitive walkthrough is only as good as the expert performing it"*. Furthermore, this technique has a narrow focus that may be useful for certain types of system but not for others [4].

The cognitive walkthrough technique takes longer time than the heuristic evaluation for evaluating the same part of, for example, a website, since it examines each step of a task. Hence, cognitive walkthrough is a useful technique for examining a small part of a system in detail, whereas heuristic evaluation is useful for examining whole or parts of a system [4].

Table 6. Some advantages and disadvantages of three evaluation methods, author's interpretation from Preece et al. [7].

Method	Advantages	Disadvantages
Usability Testing	Identifies serious and recurring problems Avoids low-priority problems	Requires user interface experience High cost Misses consistency problems
Heuristic evaluation	Identifies many problems. Identifies very serious problems Low costs	Requires user interface experience Requires several evaluators
Cognitive Walkthrough	Helps define users' goals and assumptions. Can be used by software developers	Needs task definition methodology Misses general and recurring problems

The GOMS Evaluation Model

This is a generic term used to refer to a family of models that vary in their granularity as to what aspects of a user's performance they model and make predictions about. These include the most effective strategies to use when performing different tasks and the time it takes to perform these tasks. The models are used mainly to predict user performance when comparing different applications and devices. Two of the most well-known members of the GOMS family are the GOMS model and the keystroke level model which both will be described below [4].

The GOMS model

The GOMS model was developed by Card et al., in the early eighties in an attempt to model the knowledge and cognitive processes involved when users interact with systems [7].

The name GOMS is an acronym which stands for goals, operators, methods and selection rules and consists of descriptions of the methods needed to accomplish specified goals. Goals refer to a particular task that the user wants to achieve and operators refer to the cognitive processes and the physical actions that need to be performed in order to achieve those goals. The difference between a goal and an operator is that a goal is achieved and an operator is executed [4].

The methods are a series of steps consisting of actions that the user has to perform to accomplishing the goals [4]. Selection rules will, depending on the context, choose the appropriate method to accomplish a goal when there is more than one method available [7]. GOMS models may be constructed during the design or after the implementation of a system, but, as with all models, it must be appropriate for its purpose. Kieras suggests that some of these purposes could be; predicting human performance with a design, producing an evaluation of the consistency, completeness, naturalness and efficiency of the design and providing suggestions for improving the design [7].
Once the GOMS model analysis has been completed it can be used in a number of ways, for example

- Checking for consistency of methods.
- Predicting the quality of an existing system or a prototype.
- As a quantitative evaluation technique.
- Checking that most frequent goals are achieved by relatively quick methods.
- Choosing between alternative designs [7].

One of the main benefits of the GOMS model is that it allows the developer to perform comparative analyses for different interfaces or computer systems relatively easily. While the GOMS model can be a useful help when making decisions about the effectiveness of a new product, it is not often used for evaluation purposes. Part of the problem is its highly limited scope, for example, it is only intended to be used to predict expert performance, and does not allow for errors to be modelled. This makes it much more difficult to predict how an average user will carry out their tasks when using a range of systems. In most situations, it may not be possible to predict how users are going to perform. Many unpredictable factors come into play including individual differences among users, learning effects, mental workload, fatigue and social and organizational factors [4].

The Keystroke level model

The keystroke level model differs from the GOMS model in that it provides actual numerical predictions of the user performance. It enables the designer to predict the time it will take a user to perform a task using the system [1]. The main benefit of these predictions is that different features of systems and applications can easily be compared to see which might be the most effective for performing a specific task [4].

This model is simple but effective. The central idea behind it is that the time for an expert to do a task on an interactive system is determined by the time it takes to make the keystrokes. So, to get the total time for the execution of a task, the method for the task has to be written down, the number of keystrokes required has to be counted, and finally multiply by the time per keystroke. However, to give a true picture, operations other than keystrokes must be added to the model [1].

The keystroke-level model has several restrictions like; the user must be an expert, the task must be a routine task, the method must be specified in detail and the performance must be error-free. These restrictions are important and must be carefully considered when using the model. However, the keystroke-level model represents an appropriate idealisation of this aspect of performance and that it is a flexible tool allowing the system designer to deal systematically with this aspect of behaviour [1].

Fitts' Law

Fitts' Law can in addition to the GOMS model and the keystroke-level model also be used to predict expert, error-free performance for certain kinds of tasks. It is used to predict the time it takes to reach a target using a pointing device, taking into account the size of the object and the distance to it. Specifically, it is used to model the time it takes to use a mouse, or other input devices, to click on objects on a screen. The conclusion is that the bigger the target is the easier and quicker it is to reach it. One of the main benefits with this law is that it can help designers decide where to locate for example buttons, what size they should have and how close together they should be on a screen display. Fitts' law also predicts that the targets that are most quickly accessed on any computer display are the four corners of the screen [4].

So, which of all these methods described above is the best? Well, there probably is no such thing as the best method. It depends on what the developer is looking for, how much money that can be afforded to be spent and what is going to be done with the results collected. The only conclusion that can be made is that the different methods are good for different things hence, they tend to find different classes of usability problems [10].

The table below illustrates some of the characteristics of the three main evaluation paradigms described above, which may be a help when choosing between the different methods [4]. In addition, the field study paradigm described in Section 2.4.1, is also included in this table. Here it is thought of as being an evaluation paradigm. Keep in mind that the predictive paradigm contains the heuristic evaluation, the cognitive walkthrough, the GOMS model, the keystroke level model and Fitts' law.

Table 7. Characteristics of different evaluation paradigms, author's interpretation from Preece et al. [4].

Evaluation Paradigms	**"Quick and dirty"**	**Usability testing**	**Field studies**	**Predictive**
Role of users	Natural behaviour.	To carry out set tasks.	Natural behaviour.	Users generally not involved.
Who controls	Evaluators take minimum control.	Evaluators strongly in control.	Evaluators try to develop relationships with users.	Expert evaluators.
Location	Natural environment or laboratory.	Laboratory.	Natural environment.	Laboratory-oriented but often happens on customer's premises.
When used	Any time you want to get feedback about a design quickly. Techniques from other evaluation paradigms can be used-e.g., experts review software.	With a prototype or product.	Most often used early in design to check that users' needs are being met or to assess problems or design opportunities.	Expert reviews (often done by consultants) with a prototype, but can occur at any time. Models are used to assess specific aspects of a potential design.

Type of data	Usually qualitative, informal descriptions.	Quantitative. Sometimes statistically validated. Users' opinions collected by questionnaire or interview.	Qualitative descriptions often accompanied with sketches, scenarios, quotes, other artefacts.	List of problems from expert reviews. Quantitative Figures from model, e.g., how long it takes to perform a task using two designs.
Fed back into design by…	Sketches, quotes, descriptive report.	Report of performance measures, errors etc. Findings provide a benchmark for future versions.	Descriptions that include quotes. sketches, anecdotes, and sometimes time logs.	Reviewers provide a list of problems, often with suggested solutions. Times calculated from models are given to designers.
Philosophy	User-centered, highly practical approach.	Applied approach based on experimentation, i.e., usability engineering.	May be objective observation or ethnographic.	Practical heuristics and practitioner expertise underpin expert reviews. Theory underpins models.

2.4.5 Implementation

The implementation of the software should be a continuous and escalating process running in semi-parallel with the earlier phases to build up a fully functional application. It is important to include the results from the evaluations throughout the process and to take care of new problems that arise during the implementation [18].

When the design has been around the prototyping cycle enough times for the designer to feel confident that it fits requirements, it is time to go over to the implementing cycle to be able to evaluate and possibly audit the finished system [4].

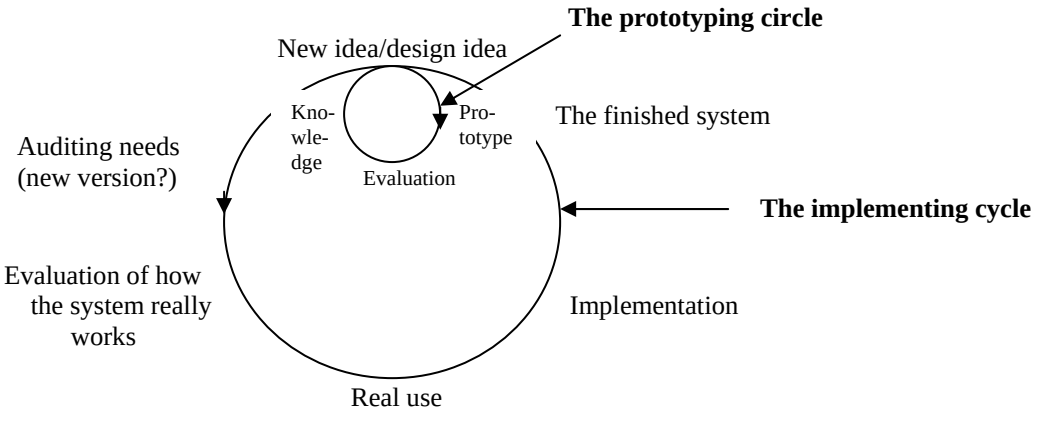


Figure 11. The implementing cycle,[28] author's interpretation.

[28] http://www.eat.lth.se/Kurs/Material/MAM061

3. Method

In order to create applications that are going to be useful to the users, the users need to be involved throughout the whole design process to evaluate the designs so the designer can make sure that the users' needs and expectations are being fulfilled. This is why the lifecycle model for interaction design developed by Preece el al., described in Section 2.3, which is based on an iterative design process, has been used as a ground for this thesis and project. The next chapter is divided into the following main sections:

Identifying needs, establish requirements and evaluation of these

Benchmarking and brainstorming

These two methods were used to enable a fast generation of ideas about what information and what services that were going to be offered on the applications and how they were going to be designed.

User profiling, task analysis and field study

In addition to the brainstorming sessions and the benchmarking, a user study and a field study were performed, to help with the user profiling and the task analysis.

Goal analysis

With help from the results, tips and comments from the previous phases an overall picture of the project could be created and usability goals and user experience goals for the project were set up.

Develop alternative designs, prototype, user test and evaluate

Once the set of requirements and goals were established they were used to guide the design process.

Testing the menu structure for the homepage

The menu structure for the homepage was tested to be able to see if the right function had been placed under the right category with the right headline. The test was performed on two different kinds of prototypes.

Conceptual and physical design, prototype and evaluate

A couple of different design alternatives were raised, to see how the information and the functions on the applications should be structured so the overview, the orientation and the navigation was going to be facilitated. These were then tested on the intended users with help of prototypes and the results were then evaluated to see if the designs lived up to the goals that were set for them in the beginning of the project. If not, they were redesigned, tested and evaluated again. A lot of different evaluation methods and design guidelines that were described in chapter 2 were used.

Implementation

When the designs fulfilled the requirements and goals; set up for the applications in the beginning of the development, it was time to leave the prototyping cycle and move on to the implementing cycle to be able to evaluate, and if needed, audit the finished system. A last user test was performed and the results of this showed that the users were very pleased with the applications and did not have any further wishes or complaints.

Developing tools
Front Page was used to develop the applications and an access database was used to hold the information about the arriving vessels. Programming languages used were JavaScript, VBScript, ASP and HTML.

4. The design iterations and the implementation of the applications

4.1 Identifying needs, establish requirements and evaluation of these

4.1.1 Benchmarking and brainstorming

To be able to create a vision for the homepage, ideas about what information and what services that were going to be offered on it were needed. This was done by a benchmarking study to find out what kind of information, services and functions the other ports in Sweden offer their customers at their respective homepages. All their headlines and functions were written down on pieces of paper. It was kind of like doing brainstorming, see Section 2.4.1, but instead of doing it with people as usual, competitors' homepages were used. However, a real brainstorming session was performed as well with the managing director, MD, at SSHAB and all the pieces of paper from both this brainstorming session and the benchmarking study were then put in the same heap.

This method worked very well since it enabled a fast generation of ideas about how the homepage was going to function and how it was going to be designed. After the brainstorming session was finished the results from it were gone through with the MD at SSHAB. To summarize, refine and structure the favoured ideas from the brainstorming the method function analysis was used, see Section 2.4.1. A division of the functions on the pieces of papers were performed depending on how useful they seemed to be. A function that was regarded as absolutely decisive for the system to fulfil its central purpose was classified as necessary, a function that was good but not necessary was classified as desirable, a function that did not need to be a part of the system for it to work was classified as unnecessary and was put aside. The necessary and the desirable functions were given short, clear, and unambiguous names and were sorted into related groups with main headlines which hopefully were going to be meaningful to the users. See Appendix C for the results in Swedish.

4.1.2 User profiling, task analysis and field study

In addition to the brainstorming sessions, a user study was performed, which helped when doing the user profiling, described in Section 2.4.1, i.e. to get a concrete understanding of who the users of the different applications are going to be and to capture the characteristics of those different kinds of user groups. This study also helped to do a task analysis, see Section 2.4.1, i.e. to get a better understanding of the tasks the users will be performing or will want to perform with the logistic application and with the homepage.

A part of this user study was performed as a field study, see Section 2.4.1. Two weeks time was spent at SSHAB to observe and to study the users in their natural environment to get to know them and to get a better idea about the context in which the applications will be operating, to see what kind of functions that are wanted. Information was also gathered about what problems the users are likely to encounter with the applications and what is happening in the environment in which they are going to be used. During the whole time notes were taken and interesting thoughts and ideas were written down.

An unstructured interview was also performed with five of the employees at SSHAB to question them about their experiences with the existing homepage, electing what they liked

and disliked about it, what they would like to see changed, what they want to be able to do with the new homepage, what their needs are, what kinds of services and functions they would like to have on it and what kind of information that would be of interest. A few main issues that were going to be answered were written down on a piece of paper. However, instead of following these accurately, a discussion with each of the users was conducted in order to gather as much information as possible about what they think. See Appendix D for the questions and answers, in Swedish, to this unstructured interview.

The results

The new experiences that were gathered during the user study were that the employees do not visit the existing homepage very often, if at all, since it never gets updated, hence it is enough to have seen it ones. The information is written partly in English and partly in Swedish which makes it inconsistent and the images that are used are of very poor quality. The overall impression of the existing homepage was that it is very boring. The things that would make the employees visit the homepage regularly would be if interesting information was provided and if it was kept up to date. Another thing that would make them visit the homepage more often would be if it could be used in their daily work, for example, to see what vessels that are expected to the port, when they will arrive and what cargo they are going to load respective to unload. Customers should also be able to get the information they need through the homepage instead of phoning the employees at SSHAB.

Hence, a compilation of the results from the brainstorming and the user study was that the users through the homepage fast end easy should be able to form an opinion about the port; is it a stable company, who owns it, what capacity do they have, history about the company, their business concept, safety issues, what kind of products they handle, what kind of resources they have, what kind of services they offer, prises and conditions, a table showing the arriving vessels to the port, links, contact information etc. The homepage is going to be so informative that it is going to replace all brochures about the company. Volvo and the suppliers are going to be able to log in via the homepage to get to the logistic application in order to see the status of their goods. Earlier they had to log in via a page with only a log in function on it, in order to get this information.

The applications are going to provide shortcuts to links that are going to be used often by frequent users. These shortcuts are going to consist of icons placed at a visible, easy to reach place. The homepage is in addition to Swedish, going to be available in English since the main part of the suppliers and customers are international. The logistic application is only going to be available in English.

An idea that came up during the brainstorming session about having a search function on the homepage is not going to be implemented in the design since the structure of the information on the page is going to be so clear that this function is not going to be needed. If the function is asked for later on it can be added in a later version of the homepage. Another idea from the brainstorming session that is not going to be implemented is to have a loading page that is displayed while the applications are loaded. This idea was eliminated since having a loading page can be very irritating for frequent users.

The target groups for the homepage are mainly going to be companies, existing and future customers within the transport industry, international organisations, shipping companies, ship

brokers, forwarders, competitors, coastguard, custom, the Swedish maritime administration and of course the employees at SSHAB.

The target groups for the logistic application are going to be the employees at Volvo, the suppliers and the employees at SSHAB. The suppliers are only going to have access to their own coils while the employees at SSHAB and Volvo are going to have access to all the coils.

As mentioned, the applications are going to be developed for customers to SSHAB and companies etc., i.e. not so much for private persons. As a result of this, the assumption was made that the users of the applications are using broadband not a modem, which is important since they, especially the homepage, contains of a lot of images that via a modem would have taken forever to download.

To be able to design the application that is going to take care of the information about the arriving vessels to the port, which were an idea that came up during the task analysis of the homepage, another unstructured interview had to be performed. This interview was performed with two of the employees at the ship broker division at SSHAB, since they are the ones that are updating the arriving vessels list today. They do this by entering the information about the vessels in an excel document and then email it to the different divisions at SSHAB.

The reason for the unstructured interview was to question them about how the service of updating the information about the arriving vessels is going to function and what they want to be able to do with the application, what their needs are, what kind of services and functions they want to have on it etc. A couple of main questions that would have to be answered were written down on a piece of paper. However, just like the unstructured interview described above, rather than following these questions precisely a discussion with the users were carried out in order to gather as much information as possible about what their thoughts were. The unstructured interview was done on both persons at the same time. See Appendix E for the questions and answers, in Swedish, to this unstructured interview.

The new experiences that were received in this user study were that everyone that visits SSHAB’s homepage should be able to see which vessels that are expected to the port, which port they are coming from, what date and time they are arriving respective departing. They are also going to be able to see what type of cargo the arriving vessels are going to load respective unload. This information is going to be sorted by date and time of arrival and is going to be available in a table accessed through the homepage. Next to the table, a login possibility is going to be provided for the authorised personnel, like the forwarders, the ship brokers and the administration, where they can log in when they want to get more detailed information about the vessels, see Figure 12.

Figure 12. Table of the arriving vessels, shown to all users.

When the forwarders and the administration log in, they are going to see a table that is going to show additional information like p-number[29], quantities and possible remarks, see Figure 13. This information is not shown to all the users due to security reasons.

Figure 13. Table of the arriving vessels, shown only to the forwarders and the administration.

However, when the ship brokers log in, a new application will be displayed that is going to hold the same table that the forwarders and administration is going to see but in addition, it will give them the authority to add, delete or change the vessel information in this table, see Appendix F for a picture of this application. For this application a database must be created that constantly is going to be updated with the latest information concerning the vessels. These updates must be easy and flexible to do since they are going to be done several times a day depending on the weather, wind and the traffic situation. A form, which is going to be described in detail further down, needs to be created.

The vessel information disappears automatically from the table the time and date the vessels depart from the port.

[29] Reference number.

4.1.3 Goal analysis

With help from the results, tips and comments from the brainstorming and the user study an overall picture of the project could be created. Usability goals for the project were set up in terms of ”REAL” and other terms mentioned in Section 2.4.1.

When it comes to the relevance of the applications, i.e. how well they serve users’ needs, it is going to be high. The services and functions on the applications and the information that is going to be shown on the interfaces should be relevant for the tasks the users are going to perform. It should be easy for the users to know how to interact with the applications which should not contain any unnecessary information.

The efficiency of the applications can be divided into two different efficiency measures; how efficiently users can carry out their tasks through a minimal number of steps using the system and how often the users make mistakes, the error rate. Since the relevance for the applications is going to be high and there are not going to be any unnecessary or confusing functions, the users should quick and easy, with very few mistakes, be able to manoeuvre the applications and to find what they want.

When it comes to the users’ attitude, i.e. their subjective feelings towards the applications, it should be positive since a lot of the employees’ and the customers’ daily tasks will be facilitated by being done through the applications.

The applications learnability can be measured in two different ways. The first way is to measure how easy the applications are to learn for initial use. It is going to be easy for the users to find the information they are looking for and no training is going to be needed. The menus should be structured in a way so that they have a natural grouping of the different alternatives and if the users have gone through the alternatives once, they should remember where they were to the next time they are going to use it. The form, used by the ship brokers, for filling in the information about the arriving vessels is also going to be easy and intuitive to use. Consequently the learnability for all the applications is going to be very good. A novice user is going to be able to learn how to use the homepage in under ten minutes.

The second way of measuring the learnability of the applications is to measure how well the users will remember their skill over time. Since the applications relevance is going to be very high, the users are going to remember where the information is, and how to use the applications from time to time, and if not, it is not going to take long time for them to learn this again.

The effectiveness i.e. how good the system supports its tasks, is going to be very good. The applications are going to support the tasks that are needed by the users.

The applications are also going to be safe for the users to use, i.e. the system is going to protect them from dangerous and undesirable situations.

The utility for the system is also going to be good i.e. the applications are going to provide the right kind of functionality.

In short, the usability goals and hence the usability requirements, see Section 2.4.1, for the applications are that the information on them are going to be relevant, efficient, easy to learn and remember and are going to provide the right kind of functionality. They are also going to

be effective and safe to interact with and outputs from them are going to be clear and unambiguous.

The user experience goals for the applications are that they are going to be enjoyable, satisfying, motivating, helpful and rewarding to interact with. They are also going to be emotionally fulfilling and aesthetically pleasing.

4.2 Develop alternative designs, prototype and evaluate

Once the set of requirements were established they had to be organized in a suitable way and be given a suitable structure. This was done by transforming them into a conceptual model which, as described in Section 2.4.2, describes what the product should do, behave and look like. At this point there were no considerations taken to possible technical restrictions.

As mentioned in Section 2.4.2, it is important that the users know how to carry out their tasks using the system and that they can predict the outcome of their actions, i.e. the gulfs need to be bridged, see Figure 6 in Section 2.4.2. For this to be achieved the system must make things visible hence provide the necessary data and information so the users can carry out their tasks. This was done by structuring the information that was going to be a part of the interface, in a logical way. The first thing that was done was to create the menu structure for the homepage. Since the other applications are going to be accessed through the homepage it needs to be clear and comprehensible to the users. Care was taken to match user expectations when choosing the layout of the menu, the grouping and the order of the menu items.

4.2.1 Testing the menu structure for the homepage

To be able to see if the right function was placed under the right category with the right headline, a lo-fi prototype, see Section 2.4.3, made out of paper was created. This prototype was tested on the MD with the evaluation method ”quick and dirty”, described in Section 2.4.4, to get quick feedback. Every time he pressed a “link” on the paper, the images were changed in about the same way it would work in reality. Unfortunately a paper prototype was not the right choice of prototype here since it became very complicated with a lot of paper changes and so on, hence the essential part of the test, i.e. to examine the menu structure, did not really come through.

Instead a simple hi-fi prototype, described in Section 2.4.3, created in FrontPage was used to test if the categorization of the links and the order they were presented in worked. This prototype only contained a menu structure, see Figure 14, and was tested on four of the employees at SSHAB, who were all randomly chosen. The test was conducted in a laboratory-like condition and contained four different questions that were going to be answered by the users with help of the prototype. Each test person was first given a short description of the product and possible questions were answered. It was stressed that the prototype used in the test only was a prototype that was going to test the menu structure so the test persons would not comment on the layout, typeface or the choice of colour.

Figure 14. The old menu structure when the user had chosen the headline “About the port” and then the headline “Port facts”. The blue colour indicates that the headline has links beneath it.

The questions that were going to be answered were chosen because they considered the management of the menu in a good way and covered the important parts that needed to be tested. With help of these questions potential faults in the design of the menu structure were hopefully going to be found. A pilot study of the questions were done before they were used in the user test, to make sure that they really were clear and that they tested the menu structure in a good way.

The test persons were requested to think out loud during the whole test and to comment every step in their way of thinking. During the whole test period notes were taken over the users’ thoughts and other observations that were made. Shneiderman’s five measurable human factors, described in Section 2.4.4, were kept in mind during the test and were used later on to evaluate the design. After the test the users gave their opinion about how they perceived the menu structure, if they thought something was hard to understand etc. Spontaneous comments were also given. See Appendix G for questions and answers, in Swedish, to the user test.

A compilation of the results from the user tests were that the users thought that the overall structure of the menu was great and that the right links were put beneath the right headlines. But all the users made the comment that the headline ”Vessel operations” beneath ”Services” should be named ”Goods handling” instead since goods are being delivered to the port both by train, truck and vessel. Other comments that were given was that the most relevant option should be placed first to avoid frequent long scans and scrolls. The suggested order in which

the headlines beneath the main headline “About the port” should be in is the one presented in Figure 15.

Figure 15. The new menu structure when the user had chosen the headline “About the port” and then the headline “Port facts”. The blue colour indicates that the headline has links beneath it

Another comment was that the headlines only should have a capital letter in the beginning of them. Yet another comment was that the link ”Organisation scheme” beneath the headline “Organisation” was unnecessary. This information should instead be presented on the page ”Contact us”. A request about an additional link called “Expansions” that were going to lie beneath the headline ”Future” was also made.

Hence the results from the user test using Shneiderman’s five measurable human factors were as follows; it did not take the users especially long time to learn were the different functions in the menu were placed and the it hardly took any time at all for them to carry out the predetermined tasks. All the users that were tested had the same difficulties finding the right headline to answer the question about what kinds of goods that are handled by the port, but this was due to a faulty and confusing name had been given to the link, so there were no actual errors done by the users. Since the structure of the menu is easy to learn, the users will not have any problems knowing how to use it the next time they will want to use it. As mentioned above the users were happy with the overall structure of the menu except for some details mentioned above.

After the results were examined, the problem found and the proposals given during the test were taken care of. After this the menu structure was tested again to see how the change affected the users' performance. This was done by a "quick and dirty" evaluation with one person. And the result was that now the menu structure was completely clear and easy to use.

4.2.2 Conceptual and physical design, prototype and evaluate

If the conceptual model is clear and consistently used in the design, it is easier for users to apply what they have learnt in one part of the system, in other parts of the system as well, as described in Section 2.4.2.

The system image should make the design model clear to the users, so they get a correct understanding of the system, which in turn will make them use it effectively. As mentioned in Section 2.4.2, the arrangement of the objects on the screen determines not only how good they look but how easy they are to understand and use. A couple of different design alternatives were raised, to see how the information and the functions in the applications should be structured so the overview, the orientation and the navigation is facilitated. To visualize the different design alternatives, lo-fi paper prototypes were used which in turn were evaluated with help of the "quick and dirty" method on six persons.

They all decided that the design structure proposed by Jeffrey Veen in Section 2.4.2 that divides the interface into three main areas, see Figure 9 in Section 2.4.2, were the one that provided the best organization on the screen and structured the information and functions in the most natural way. A suggestion that came up was to add a fourth area which would contain the shortcuts that are going to facilitate for frequent users, see Appendix H. The same structure is going to be used in all three applications to add consistency to the system.

When the design structure had been decided on, it was time to design applications that would consider more concrete and detailed issues of the interface. It is important, as described in Section 2.4.2, to present the information on the interface in a way so the information as much as possible can be handled automatic since the humans' conscious level is strongly limited in capacity. This can be done by making the interface as obvious as possible to the users i.e. displaying the right amount of information at the same time and in a good way on the screen, grouping similar things together, display the information in a clear and consistent way, use icons in a correct way etc.

Interaction styles and graphic design

A good interface is built up by a set of elements that work together to produce a coherent interface that gives a sense of continuity and consistency. The different style guides, mentioned in Section 2.4.2, were used to create this feel of consistency of the interface with colour, position etc. The "keep it simple" rule, mentioned in Section 2.4.2, was kept in mind during the whole design phase.

The design of the homepage can be viewed in Appendix I, the design of the logistic application in Appendix J and the application for updating the information about the arriving vessels can be viewed in Appendix F.

Direct manipulation

The interfaces of all the three applications are built up by direct manipulation, mentioned in Section 2.4.2, so the users can carry out their tasks rapidly just by pointing at visual representations of objects and actions, and observe the results immediately.

Structural side to the interface

To organize the different areas on the screen and to create a good balance and structure, an invisible grid, windows and structural panels were used to positioning the different elements in the layout of the applications. All the applications have the same structure to create a consistency between them. They all have the same top picture and the same icons are used. They all have a visual anchor that helps the users to navigate. This visual anchor clearly, but invisibly, divides the menu part and the information part of the interface at approximate ¼ of the total width from the left. The menu in all the applications lies on the same line as the picture or the information in the information part of the applications to follow the grid.

To create a balance between the elements on the screen so that different objects do not appear to be floating, a grey-blue background colour has been used at the homepage application. It is applied on both side of the main page so the page appears to be centered on the screen

Functional side to the interface

Buttons

The buttons used on the applications all have the same layout and are clear and unambiguous hence they reveal their purpose, to be pressed, at first glance.

Menus

The menus used in the different applications differ a bit in the way they are constructed. The menus used in the logistic application and the one in the application, where information about the arriving vessels are being changed, both have a linear structure, while the menu used on the homepage is designed as a drop-down menu with three levels. This because the latter one has so many different options that it would be too hard to grasp if all the options were shown at once in a linear menu. As mentioned above, the headlines in the drop-down menu on the homepage were tested so that the right links were placed beneath the right headline, the phrasing of the menu items were right and at that they were put in the right position in the menu. This was not needed for the linear menus since these do not have that many links.

To make the drop-down menu clear and comprehensible, all the main headlines have a dark blue background colour and are written in white, see Figure 16.

Figure 16. The menu structure on the homepage.

When a headline is chosen the second level headlines are displayed. They are written in a dark blue colour on a white background. In addition to this, they are indented so it becomes really clear which headlines that are the second level headlines. Those second level headlines that have third level headlines have an arrow to their right that is pointing to the right, see Figure 17.

Figure 17. The menu structure when “About the port” has been selected.

When a second level headline like this is chosen, the colour of its background changes to a light blue colour, the arrow points down and the third level headlines are shown indented with dark blue colour on a white background, see Figure 18. This is done to make it clear what headline that is chosen and to give the users a clear idea of what will happen when they make a selection. It also helps the users to get a clear sense of progress or position within the menu. It is always possible to go back to earlier choices by just choosing another headline in the menu.

Figure 18. The menu structure when "About the port" and then "Port facts" has been selected.

The drop-down menu is limited to have three levels since, as described in Section 2.4.2, there appears to be a greater chance of users becoming lost or disoriented when the depth goes to four or five levels.

For all the three menu structures used in the different applications, the most commonly used functions were placed at the top to avoid frequent long scans and scrolls and the names were made short, clear, and unambiguous.

Icons

A couple of different icons were designed and used in the applications to increase the understanding and to facilitate for frequent users. The same icon designs were used in all three applications. Mainly the icons used were designed with existing traditions and standards in mind. However, one icon, the contact icon, was designed as an 'at'-sign instead of the usual envelop image because of the reason that it is more common to send an email than a letter these days. Two different designs of this icon were developed, one with the 'at'-sign on top of an envelope and one with only the 'at'-sign, see Figure 19. A "quick and dirty" evaluation was carried out with three persons, who all thought the 'at'-sign icon was the best since the 'envelope-at'-sign icon was not totally clear.

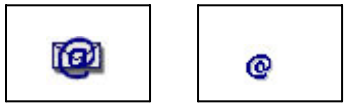


Figure 19. The 'envelope-at'-sign and the 'at'-sign icons.

The icons help to make the interface more visually appealing and they are all uniform when it comes to size, shape and colour. The icons were designed so that they would be immediately recognizable to the users but describing text was also provided beneath each of them. There have been a lot of different opinions about this. Norman claims, in Section 2.4.2, that if simple things need text to explain what they do they are not clear enough. However, as Preece et al. points out, in Section 2.4.2, if an icon has text beneath it, it becomes bigger and hence it is easier to recognise and it is accessed faster.

Form

As mentioned before, a form had to be developed that was going to be used by the ship brokers when they are going to add or change the information about the arriving vessels, see Figure 20.

Figure 20. The form used for updating the information about the arriving vessels.

This form is going to facilitate the data entry and is displayed in a separate window which will be closed when either the 'Update' or the 'Cancel' button on it is pressed. It provides the users with a display of related fields among which the users can move the cursor with the 'tab' key and enter data where desired. The field labels were carefully chosen to be understandable to the users so they will know what to enter in which fields.

There is a logical sequencing of the fields that starts at the upper left corner and goes down to the bottom right corner. It has a consistent layout with a lot of white space in between to make it easy to interact with for the users. Consistent terminology and abbreviations have been used and comprehensible error messages for unacceptable values are provided. To avoid confusion about which format to write times and dates in the users just have to choose these through the drop-down menu. The drop-down menus are also used so that the users easily can select an item from them with the pointing device and hence the possibility of keying errors is reduced.

When the fields are filled in, the user indicates the completion of the form by pressing the 'Update' button on the bottom right to update the database and the table. There is also a 'Cancel' button that will terminate the input of the form. This button is placed in the bottom left corner of the form so the users do not press this button by mistake when they are going to update the table. However, when the 'Cancel' button is pressed a confirmation box will appear which will ask the users if they really want to cancel the form.

Between the fieldnames on the left hand side and the input fields on the right hand side of the form is the red thread of the application; the visual anchor. This thin, empty space stretches through the whole window and gives a natural separation between the field names and the input fields. Overall, the layout of the form has an airy feeling with a lot of white spaces between the different input fields.

Informational side to the interface

Text

The applications speak the users' language, using words, phrases and concepts that should be familiar to them. The applications also make the information appear in a natural and logical order. The content on the pages are short and precise using headlines to capture the main points of a paragraph, so the user can scan the page for relevant information, instead of having to read it in detail. The same typeface, Arial, has been used throughout all the applications to create a sense of consistency so the users are not confused whit a lot of different typefaces and styles. Arial is a neutral, san serif typeface which is easier, than serif typefaces, to read on the screen.[30] Cascading Style Sheets[31], CSS, have been used to make sure that the headlines and the text were created in the same way with the same size and colour.

Image

The images used on the applications are all of jpeg or gif formats. Since the MDs attitude was that images say more than thousand words, there are a lot of images used especially on the homepage. But since the applications are intended mainly for users with broadband, the downloading times should not be that long. Anyway, few graphics are used on the welcoming

[30] http://www.eat.lth.se/Kurs/Material/MAM061
[31] Tool to make web pages more visually appealing [26].

page, as Nielsen suggested in Section 2.4.2, so the users can choose whether they want to visit links where there are a lot of pictures or not.

The image placed at the top of all the applications was originally created in several different styles. A “quick and dirty” evaluation was performed on a couple of different persons to see which style they preferred. The one shown at the top of the applications in Appendixes F, I and J was the one they liked the most and is the one that is going to be at the top of all the three applications. One of the persons asked, suggested to have a slogan beneath the picture but this idea was not implemented in the design since the concept of the port is going to be clear anyway. For the homepage the image at the right of the top picture will change depending on which headline that is chosen in the drop-down menu, see Figure 21.

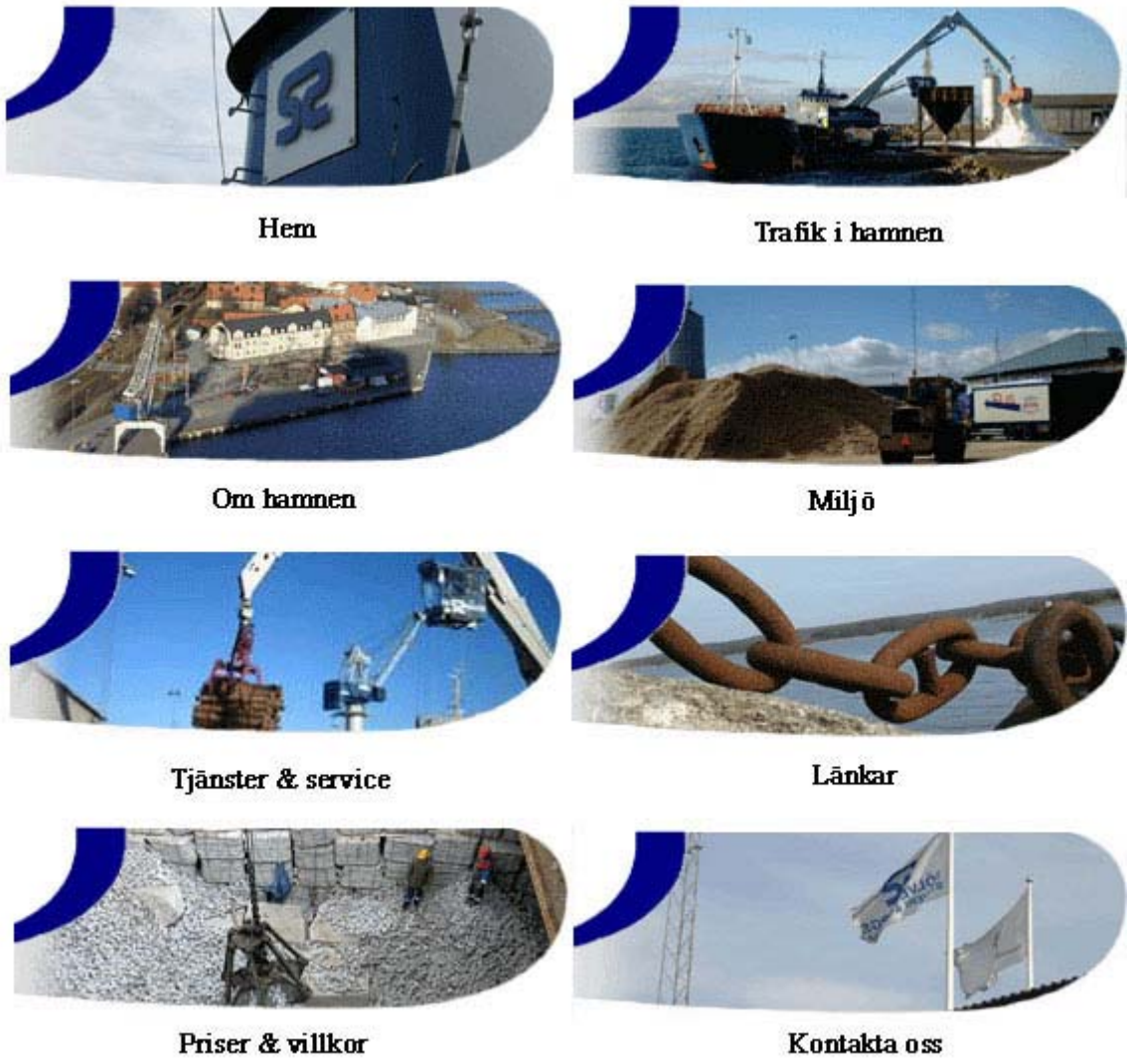


Figure 21. The top pictures to each headline in the menu for the homepage

Visual side to the interface

Colour

The interface of all the three applications uses almost exclusively different shades of blue. This colour was chosen because it is relatively neutral and not too trying for the eyes. Blue gives a distant spatial effect, see Table 4 in Section 2.4.2, which is good since the homepage is not going to be pushy. In addition, it gives a restful psychological effect, which is something that has been strived for.

The different shades of blue are used to group related items on the screen, especially this is used in the menu structure on the homepage to stress the different levels. As mentioned above, different colours can make an interface more visually appealing however a mix of too many different colours for the different parts of the application would have been confusing and would have made them hard to watch, and furthermore it would have broken the ground rule to use colours sparingly.

A closer analysis of the design

To do a closer analysis of the design and as a resource of checking for usability Shneiderman's eight golden rules of interface design, see Section 2.4.2 for details, were used.

1. Strive for consistency.
The applications seem to fulfil this rule since the menu structures, used in both the logistic application and in the application where the updating of the arriving vessel information is going to be done, were constructed in the same way. This rule is also fulfilled in the menu structure on the homepage since it is handling all the choices in the same way, where a hierarchic tree is followed. There is also a consistency in the choice of colour used throughout the applications. All the icons are blue, except for the English or the Swedish flags for obvious reasons, with blue text beneath them. At the homepage, all the main headlines are written in white text on a dark blue background and when a second level headline is chosen the background of it turns into light blue. The second level headlines are written in dark blue in all the menus used in the different applications. The buttons used in all the applications have the same colour and shape.

2. Enable frequent users to use shortcuts.
The icons were placed in a way that will make the application faster for frequent users so they will not have to go through the whole menu to search for the link they are looking for.

3. Offer informative feedback.
The users get great feedback when they place the pointer over a link at the interface since the text or the icons gets bigger and the pointer turns into a hand. Text is also shown either next to the hand or in the status field on the bottom of the window, or both, which describes where a link leads to. When a choice of a link is made, fast feedback is given either by displaying the information in the information part of the applications or in the case of the menu on the homepage, by displaying the second or third level of headlines.

4. Design dialogs to yield closure.
Since every action gives proper feedback, the users never have to wonder if their action was completed or not. Especially, the form for filling in information about the arriving vessels, fulfils this rule since every sequence of action begins with the user entering information or

choosing an option in the drop-down lists, and ends with the user pressing either the ’Update’ or the ‘Cancel’ button.

5. Offer error prevention and simple error handling.
Since the functions on the applications are relative simple, there is not much that can go wrong when handling the menus or the icons. However, if the users make the wrong choice of a link, the error can easily be fixed by simply using the menu again to make a new choice or to choose the right icon.

When it comes to the form for filling in information about the arriving vessels, the users may choose the wrong option in the drop-down menus or type the wrong information in the fields. These errors can however easily be fixed by just using the menu again and make a new choice or in case of the latter error, erase the faulty information in the field and enter the right one.

6. Permit easy reversal of actions.
The actions are easy to reverse. If the wrong alternative has been chosen in a menu the user just have to make a new choice. If it comes to the worst the users can always press the ’home’ icon to make everything look just like it did when they first got to the page. If the users happens to press the ’log in’ icon but do not have access they can just choose another link in the menu or among the icons. In the form it is just for the users to erase the information that has been entered or to choose a new alternative in the drop-down menus.

7. Support internal locus of control.
Experienced users expect the system to answer their actions and do not want to get stuck in boring sequences of filling in information, have difficulties finding the right information etc. In all three applications, the users are the ones that take the initiative to an action hence they are not reduced to a passive “button pusher”.

8. Reduce short-term memory load.
In these applications the users do not need to remember anything and the dispersal of the information over several pages has been avoided, which results in that the applications, most likely, fulfils the rule that humans in general only can handle 7±2 chunks of information at the same time in their short-term memory, see Section 2.4.2.

”Knowledge in the world and knowledge in the head”

Even terms like mapping, visibility, affordance, constraints, consistency and feedback, described in Section 2.4.2, were used to analyse the developed designs.

Mapping
To give the users a logical and intuitive sequence of action the menus were placed to the left of the interface on all three applications. This was done due to the fact that the users of western languages are conditioned to scan the screen from top left to bottom right, see Section 2.4.2. All the icons used in the applications are very clear with text beneath them so the users never have to guess what the effect of choosing them will be.

To give the users a logical and intuitive sequence of actions when they are going to fill in the form, all buttons are placed at the bottom since they are the last thing the users are going to press. A thing worth mentioning is that colours have not been used to emphasize the mapping,

as for example a green ‘Ok”-button or a red ‘Close/Cancel’-button. This would have been great for the natural mapping, where green signalling right and red signalling wrong, in the western cultures, however it was not used to avoid a lot of different colours appearing on the same application. Red, green and blue would have been too much and would simply not have had looked good together.

Visibility
Since all the guidelines for the layout, grouping, colours, icons, text etc. have been kept in mind during the whole design of the applications, their visibility are very good. The users are provided with the information needed to perform their tasks, not more. They decide themselves what information they want to see through choosing links in the menus or among the icons, hence the information shown on the applications are relevant to the users’ tasks. The menus are clearly visible at the left part of the applications and they have clear and unambiguous headlines. The icons are legible and placed at an appropriate distance from each other. Both these things facilitate the visibility and make it easier for the users to choose the right menu option or the right icon. There is only one control for each function which makes it easier for the users to predict the effects of their actions.

Affordance
For users with a little bit of computer skills, which the users are assumed to have in this case, it should be clear that the menu options and the icons on the applications can, and are meant, to be pressed. However, if the users are unsure about if something on the interface is a link, they will get extensive feedback if they place the pointer at the link, as mentioned in the third point of Shneiderman’s eight golden rules of interface design above.

It is easy to see that a button is in fact a button on the applications, hence they invites the users to press them. In the form, that is used when updating the information about the arriving vessels, drop-down menus are used and for users with computer skills the function are legible and the user is enticed to press the arrow buttons on these to be able to see the different alternatives. Users with little knowledge of computers should also be able to manage this since the arrow buttons on their own invites the users to press them. That the wanted alternative in the list is going to be chosen by clicking on it with the pointing device should not be that hard for the users to get either.

Constraints
If the users choose an incorrect option, i.e. chooses the wrong icon, menu item, drop-down alternative or enters the wrong information in the form there is no harm done since they just have to go back and redo their choices. No irreversible mistakes can be done with the applications.

Consistency
All three interfaces have a consistent look and use the same operations for selecting objects which makes the operations easier to learn and use. In the menu on the homepage the related functions are grouped and placed under appropriate categories.

Feedback
Every action has an immediate and obvious effect so the users can conclude that their action was actually performed. As mentioned in the third point of Shneiderman’s eight golden rules of interface design above, all links on the applications gives the users extensive feedback in many different ways. At first the appearance of the icons and the text in the menus are

changed when the pointing devise is pointing at them. The pointer also changes the way it looks from an arrow to a hand. Describing text about where a link leads to are shown either next to the hand or in the status bar at the bottom of the window, or both.

The drop-down menu on the homepage changes appearance through displaying respective hides the second or third level headlines. The second level headlines in the menu that have an arrow next to them have their own sub levels of headlines. The arrow is pointing to the right when the headline is not chosen and is pointing downwards when it is chosen. The second level headlines also gets a light blue coloured background when they are chosen to clearly mark where the third level of headlined belong to. This background colour disappears when they are not chosen anymore.

For the form used to update the information about the arriving vessels, feedback is given in several different ways. If the form is not correctly entered, comprehensible error messages will be provided. When the user has entered the data correctly and presses the 'Update'-button the window with the form is closed and the table is updated with the new information. If the users press the 'Cancel'-button they first get asked if they really want to cancel the form and if they do the window is closed and no changes are made to the table.

Nielsen's heuristics

The applications also fulfil Nielsen's heuristics for evaluating commercial websites, described in Section 2.4.4;

- High-quality content – only relevant information is included on the applications.
- Often updated – the content on the applications is going to be updated every day.
- Minimal download time – since it is assumed that the users of the applications are using broadband there will be no problems with long download times even though a lot of images are used.
- Ease of use – the interface on the applications meets all the requirements described above hence they are easy to use.
- Relevant to users' needs – for the same reason mentioned above, the information and functions on the interface are relevant to the users needs.
- Unique to the online medium – more or less.
- Netcentric corporate culture – yes.

User testing

After the first real designs of the three applications had been made a user test was performed on them to see how well they fulfilled the users' needs and expectations and to identify potential usability problems with them. Another reason for the test was to check that the users could navigate the systems to find the information they needed since this can be a major usability problem and also to check if the icons used were properly designed.

However, the application for updating the information about the arriving vessels was tested separately on just the employees that will be performing the updates. For this, a hi-fi prototype was used and the evaluation method used was the "quick and dirty" evaluation method. The results from this evaluation were that the application had a good design and that it worked as it should.

The user test used to evaluate the other two applications was carried out by letting the users test the designs using a hi-fi prototype and answer predetermined questions in a questionnaire. The prototype used was a simulation that looked just like the real applications are going to look like. It was made in FrontPage and ran on a personal computer.

The DECIDE framework, described in Section 2.4.4, was used to identify all the necessary steps in order to do a successful study and to help to see to that the main issues were taken care of when planning the test. Care was taken to make sure that the conditions were going to be the same for each participant.

To ensure that the questions in the questionnaire were correct formulated and comprehensible, a pilot study was carried out on the MD at SSHAB. This pilot study was also used to make sure that the usability test was well designed and that it captured the essential parts of the applications so it would be likely to be successful.

The applications were tested on five participants selected from the employees at SSHAB. The test was carried out at the employees own work desks and a questionnaire was used to collect their opinions. The questions were mainly concerning the layout and the design of the applications, the icons and to check the navigation support but also to check that the menu structure was satisfactory.

All participants were given the same information and were treated in the same way. See Appendix K for questions and answers to the user test in Swedish. The goal with this usability test was to let the users interact with the different applications on the homepage, on there own and answer the predetermined questions in a questionnaire. All these questionnaires were collected in a sealed envelope so the users' real attitudes towards the applications would be captured.

When the usability tests had been performed, all the questionnaires were analysed to see if there were any more changes that needed to be made to the applications.

Conclusions of the user test

Below a conclusion of the results from the user test and the "quick and dirty" evaluation of the applications are shown. These were compared to the usability goals and the user experience goals, which were set for the applications in the beginning of this project, to make sure that these goals were all fulfilled.

Relevance: Over all, the users thought the applications supported their needs very well. The applications provide the users with the information needed for them to be able to carry out their tasks and there are no unnecessary information displayed on the screens that confuses them. This indicates that the relevance is high and hence the usability goal concerning the relevance is fulfilled. A 'log in' icon shaped as a key is provided on the upper left of the interface but the only ones that have to use this are the employees at Volvo, the suppliers and the employees at SSHAB.

Efficiency: Since the relevance is high and there are no unnecessary or confusing functions, the users can quick and easy, with a low error rate, manoeuvre the applications and find what they are looking for, hence the usability goal that were set up for the efficiency is also fulfilled.

Attitude: The users thought the new applications would make their daily tasks easier, hence the usability gaol that were set up for the attitude towards the system is also fulfilled.

Learnability: The learnability for the applications was shown to be high which were put as a usability goal. Since there are no confusing information or functions and not much for the users to keep in mind, they should be able to learn how to use the applications at a first introduction to them and they should be able to remember how to use them to the next time they are going to use them, and if not, it will not take long time for the users to learn this again.

Effectiveness: The users thought the applications supported the tasks they were performing in a very good way, hence the effectiveness goal that was set up for the applications is fulfilled.

Safety: The users thought the applications were safe to use, i.e. it did not put them in undesirable or dangerous situations.

Utility: The utility for the system also appeared to be very good since the applications provided the right kind of functionality.

The user experience goals that were set up for the applications in the beginning appeared to be fulfilled as well, since the users thought that the applications were enjoyable, satisfying, emotionally fulfilling, aesthetically pleasing, motivating, helpful and rewarding to interact with.

Hence the results of the evaluation of the test showed that all the usability goals and all the user experience goals decided upon in the beginning of the project were fulfilled. However, a technical concern was brought up during the test. This was that for users with low resolution on their screens, the background should not take as much space as for users with high resolution. The icons will also have to be placed accordingly, especially the English flag since the international users should not have to scroll the page to see it. Another concern brought up during the user test was that the quality picture at the bottom left on the homepage should not be there since it was an old picture that was not used anymore. These concerns were taken care of immediately and the quality link was placed in the menu beneath the headline “About the port” instead.

Since the designs seemed to fit the requirements and goals set up for the applications in the beginning of the development it was now time to leave the prototyping cycle, described in Section 2.4.3, and move on to the implementing cycle, described in Section 2.4.5, to be able to evaluate and possibly audit the finished system.

4.3 Implementation

For the evaluation of the finished product, a last user test was performed. There were a total of eight employees participating in the test and each of them were told to interact with a hi-fi prototype of the applications and to give their last comments on them before they are going to be put into use. The prototype used was a simulation that looked just like the real applications are going to look like. It was made in FrontPage and ran on a personal computer. The test was carried out at the employees own work desks and an unstructured interview was used to collect their opinions. See Appendix L for the questions and answers, in Swedish, to the last user test.

The results showed that the users were very pleased with the applications and they did not have any further wishes or complaints.

5 Conclusion

The result of this thesis is three web applications that are going to work both as an important source of information, which will facilitate for the employees at SSHAB, but also to facilitate the information torrent between the steel suppliers in Europe, SSHAB and Volvo.

The users found the developed applications enjoyable, satisfying, emotionally fulfilling, aesthetically pleasing, motivating, helpful and rewarding to interact with. They thought they provided them with the right kind of information needed for them to be able to carry out their tasks quick and easy, with a low error rate and they did not think that there was any unnecessary information displayed on the screens that confused them. They also thought the applications were easy to learn at a first introduction and that they were safe to interact with. Hence all the usability goals and user experience goals that were set up for the applications in the beginning of this project were fulfilled.

However, even though the web applications are fully developed they are not up and running yet. A further development will be to see to that this is taken care of so the web applications can start to facilitate the torrent of information for the port operations.

6 References

[1] Preece, J. & Keller, L. (1990)
Human-computer interaction: selected readings: a reader,
Prentice Hall International (UK) Ltd
ISBN: 0-13-444910-X

[2] Helander, M., Landauer, T. & Prabhu, P. (1997)
Handbook of Human-Computer Interaction, 2nd Ed.,
North-Holland
ISBN: 0-444-81876-6

[3] Löwgren, J. & Stolterman, E. (1998)
Design av informationsteknik – materialet utan egenskaper,
Lund: Studentlitteratur
ISBN: 91-44-00681-0

[4] Preece, J., Rogers, Y. & Sharp, H. (2002)
Interaction design: Beyond human-computer interaction,
John Wiley & Sons, Inc.
ISBN: 0-471-49278-7

[5] Norman, D. (1993)
Things That Make Us Smart,
Reading MA: Addison-Wesley
ISBN: 0-201-58129-9

[6] Norman, D. & Draper, S. (1986)
User Centered System Design - New Perspectives on Human-Computer Interaction,
Lawrence Erlbaum Associates, Inc., Publishers
ISBN: 0-89859-872-9

[7] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carry, T. (1994)
Human-Computer Interaction,
Wokingham, UK: Addison-Wesley
ISBN: 0-201-62769-8

[8] Carroll, J. (2001)
Human-Computer Interaction in the New Millennium
New York: Addison-Wesley
ISBN: 0-201-70447-1

[9] Norman, D. (1990)
The Design of Everyday Things,
London: MIT Press
ISBN: 0-385-26774-6

[10] Löwgren, J. (1993)
Human-Computer Interaction – what every system developer should know,
Lund: Studentlitteratur
ISBN: 91-44-39651-1

[11] Shneiderman, B. (1998)
Designing the User Interface, 3rd Ed.,
Addison-Wesley
ISBN: 0-201-69497-2

[12] Shneiderman, B. (1992)
Designing the User Interface - Strategies for Effective Human-Computer Interaction
2nd Ed.,
Addison-Wesley Publishing Company, Inc.
ISBN: 0-201-57286-9

[13] Faulkner, X. (2000)
Usability engineering,
New York: Palgrave Houndmills
ISBN 0-333-77321-7

[14] Tognazzini, B. (1992)
TOG on Interface, 5th Printing,
Addison-Wesley Publishing Company, Inc.
ISBN: 0-201-60842-1

[15] Cooper, A. (1999)
The inmates are running the asylum,
Indianapolis: SAMS
ISBN: 0-672-31649-8

[16] Brenda, L. (1990)
The art of Human-Computer Interface Design,
Addison-Wesley Publishing Company, Inc.
ISBN 0-201-51797-3

[17] Kristof, R. & Satran, A. (1995)
Interactivity by design : creating & communicating with new media
Mountain View: Adobe press
ISBN: 1-56830-221-5

[18] Rettig, M. (1994)
Prototyping for tiny fingers. *Communications of the ACM*, 37(4), 21-27.

[19] Norman, D. (1992)
Turn Signals are the Facial Expressions of Automobiles,
Reading: Addison-Wesley
ISBN: 0-201-62236-X

[20] Schneider, W. (1993)
Människan Datateknik Arbetsliv,
Stockholm: Publica
ISBN: 91-38-92226-6

[21] Mullet, K. & Sano, D. (1995)
Designing Visual Interfaces,
SunSoft Press, Prentice Hall, PTR
ISBN: 0-13-303389-9

[22] Veen, J. (2001)
The Art and Science of Web Design,
Indianapolis: New Riders Publishing
ISBN: 0-7897-2370-0

[23] Dumas, J. & Redish, J. (1999)
The inmates are running the asylum,
Indianapolis: SAMS
ISBN: 0-672-31649-8

[24] Dumas, J. & Redish, J. (1999)
A Practical Guide to Usability Testing,
Exeter, UK: Intellect
ISBN: 1-84150-020-8

[25] Nielsen, J. & MACK, R. (1994)
Usability Inspection Methods,
New York: John Wiley & Sons
ISBN: 0-471-01877-5

[26] Deitel, H., Deitel, P. & Nieto, T. (2002)
Internet & World Wide Web – How to program 2nd Ed.,
Upper Saddle River, New Jersey: Prentice Hall, Inc.
ISBN: 0-13-030897-8

Appendix

Appendix A – The old logistic application

SÖLVESBORGS STUVERI & HAMN AB

TEL: +46 456-422 40 FAX: +46 456-127 61 E-mail: info@stuverihamn.se

Sölvesborg's harbour is strategically located in a region trading with Russia, the Baltic States, and the rest of Europe. Business has grown very positively during the past several years.

MD Jan Olsson
Ph. +46 456-422 40
Mob. ph. +46 70-529 45 79

Updated: 2004-03-01

Best resolution:
800*600

Appendix C – Results of the brainstorming and function analysis

Nödvändiga

Om hamnen:

Hamnfakta:	Kartor, Kapacitet, Hamnen i siffror, Hämta/lämna gods
Bildgalleri	
Organisation:	Ägarstruktur, Kunder, Organisationsschema
Idag	
Framtid:	Affärsidé, Visioner & mål
Säkerhet	
Historik	

Tjänster & service:

Fartygsoperationer:	Bulkgods, Skogsprodukter, Containerhantering, Coilshotell, Övrigt gods
Resurser:	Maskiner, Kranar, Våg, Bogsering
Terminal & lagringstjänster	
Övriga tjänster:	Verkstad, Kranuthyrning, Truckuthyrning

Priser & villkor:

- Öppettider
- Hamnavgifter
- Hanteringspriser
- Ansvarsbestämmelser

Trafik i hamnen
Miljö
Länkar
Kontakta oss

”Logga in” möjlighet:	För de anställda vid Volvo och SSHAB och för leverantörerna så de kan se deras produkter.
Engelsk flagga:	Det ska gå att få hemsidan på Engelska.

Önskvärda

Genvägar för frekventa användare i form av ikoner.
Tredimensionell vy av hamnen.

Onödiga

Sökfunktion.
”Laddnings sida”.

Appendix D – Questions and answers to the unstructured interview

Frågor:

1. Hur ofta använder du den nuvarande hemsidan?
2. Vad är dina erfarenheter av den, dvs. vad gillar du respektive ogillar du med den?
3. Vad skulle få dig att gå in och titta på den mer regelbundet?
4. Vilka funktioner och tjänster skulle du vilja ha på den nya hemsidan?

Svar:

Person 1
1. Aldrig.
2. Kass, ingen relevant information.
3. Om den praktiskt kan användas i dagliga arbetet och om den har uppdaterad information.
4. Se väntade fartyg, kunna logga in via hemsidan för att t.ex. se vila coils som finns i lager.

Person 2
1. Aldrig.
2. Dålig med gammal information.
3. Om informationen på den är intressant och förändras.
4. Bra information så kunder slipper ringa och fråga.

Person 3
1. Sällan.
2. Tråkig, uppdateras aldrig.
3. Om informationen på sidan uppdateras ständigt.
4. Snabbt och enkelt kunna ta reda på telefonnummer och mail adresser till de man vill kontakta.

Person 4
1. Sällan – vet allt som står redan.
2. Väldigt enkel och trist med dålig information som aldrig uppdateras.
3. Intressant information som uppdateras ofta.
4. Besökarna till hemsidan ska enkelt kunna ta reda på hur hamnen ser ut, vilka möjligheter hamnen har att hantera deras gods m.m. Vill ha en hemsida som inte bara listar information. Den ska kunna användas i det dagliga arbetet.

Person 5
1. Sällan, har ingen praktisk nytta av sidan.
2. Dålig kvalitet på bilder, blandar både engelska och svenska på samma sida.
3. Uppdaterad, bra information och att praktiskt kunna använda den i det dagliga arbetet.
4. Kunna se vilka fartyg som är väntade till hamn, när de kommer och vilka gods de ska lossa eller lasta.

Övriga kommentarer var att hemsidan främst ska rikta sig till företag, nuvarande och blivande kunder inom transport industrin, internationella organisationer, mäkleriet, speditörer, konkurrenter, kustbevakningen, tullen, sjöfartsverket, de anställda på SSHAB m.fl. Logistik applikationen ska rikta sig till de anställda på Volvo och SSHAB och till leverantörerna.

Appendix E – Questions and answers to the unstructured interview concerning the arriving vessels

Frågor:

- **Vilka ska kunna se informationen?**
- **Vad ska kunna göras med applikationen?**
- **Vilka funktioner ska finnas på den?**
- **Vem ska kunna utföra uppdateringarna?**
- **Hur ska uppdateringen av informationen ske?**

Svar:

- Alla som besöker hemsidan ska kunna se vilka fartyg som är väntade till hamnen, vilken hamn de kommer ifrån, vilken tid och vilket datum de kommer till respektive lämnar hamnen. De ska även kunna se vilket/vilka godsslag fartygen ska lasta respektive lossa.
- Tabellen ska vara sorterad efter datum och tid då fartygen ska anlända till hamnen.
- Behöriga användare såsom speditörerna, mäkleriet och administrationen ska kunna logga in för att få mer detaljerad beskrivning om fartygen. Denna extra information är p-nummer, kvantiteter och eventuella anmärkningar och visas inte för alla pga. säkerhets skäl.
- En databas med information om fartygen måste skapas och måste uppdateras ständigt. Uppdateringarna måste vara enkla att göra eftersom de kommer att göras flera gånger per dag beroende av väder, vind och trafik situationer. De ska göras via ett formulär.
- Det är bara mäkleriet som ska kunna uppdatera databasen och därmed tabellen. De ska kunna lägga till nya fartyg, ändra befintlig information eller ta bort fartygsinformation.
- Fartygen ska automatiskt försvinna från tabellen det datum och den tid då de avgått från hamnen.

Appendix F – The application for updating the information about the arriving vessels

Appendix G – Questions and answers to the test of the menu structure

Frågor:

1. **Vilka typer av gods hanteras av hamnen?**
2. **Vem äger hamnen?**
3. **Vilken kapacitet har hamnen när det gäller att lossa gods?**
4. **Vilka öppettider har hamnen?**

Svar:

Person 1
l: Personen gick först in under rubriken ”Tjänster & Service” men tyckte inte att någon av underrubrikerna passade in för det han sökte så då gick han in under rubriken ”Om Hamnen” istället. När personen då inte hittade riktigt vad han letade efter blev han lite förvirrad. Han gick in under rubriken ”Tjänster & Service” igen och med viss tveksamhet gick han in under rubriken ”Fartygsoperationer” och mycket riktigt låg det han sökte där. Hans kommentar var att ”Fartygsoperationer” var ett vilseledande namn eftersom gods även anländer till hamnen via tåg och lastbil. Han tyckte ”Godshantering” skulle vara ett bättre namn.

2: Personen gick in under rubriken ”Om Hamnen”, tyckte att rätt svar kunde finnas antingen under ”Hamnfakta” eller under ”Organisationen”. När han testade båda så såg han att det låg under ”Organisationen” med kommentaren att där låg den bra.

3: Personen gick in under rubriken ”Tjänster & Service” och hittade det rätta svaret under ”Resurser”.

4: Personen gick först in under rubriken ”Om Hamnen” men när han inte hittade det han sökte direkt gick han istället in under ”Priser & Villkor” där han hittade det rätta svaret direkt. Personen kommenterade att länken låg bra där.

Person 2
l: Personen gick in under rubriken ”Tjänster & Service” och valde sedan med tveksamhet ”Fartygsoperationer” och hittade det han sökte, dock med kommentaren att gods även kommer in med även lastbil och tåg så ”Fartygsoperationer” kanske inte var rätt namn till detta.

2: Personen gick in under rubriken ”Om Hamnen” och hittade rätt under ”Organisation”.

3: Personen gick in under rubriken ”Tjänster & Service” och sen in under ”Resurser” och hittade det han sökte.

4: Personen gav kommentaren att svaret till frågan antingen kunde ligga antingen under ”Om Hamnen” eller under ”Priser & Villkor”. Han testade ”Om Hamnen” först men när han inte hittade svaret där gick han in under ”Priser & Villkor” svaret hittades. Personen kommenterade att länken låg rätt eftersom det är ett villkor.

Person 3
l: Personen gick in under rubriken ”Tjänster & Service” och valde sedan ”Fartygsoperationer” och hittade det han sökte, dock med kommentaren att ”Godshantering” hade varit ett bättre namn.

2: Personen gick in under rubriken ”Om Hamnen” och sedan tyckte han att svaret kunde ligga antingen under ”Hamnfakta” eller ”Organisation”.

3: Personen gick in under ”Tjänster & Service” och sen in under ”Resurser” och hittade det han sökte.

4: Personen gick in under ”Priser & Villkor” och hittade svaret till frågan direkt.

Person 4
l: Personen gick in under rubriken ”Tjänster & Service” men hittade inte vad han sökte. Han gick in under ”Om Hamnen” istället och valde ”Hamnfakta”. Då han inte hittade svaret här gick han tillbaka till ”Tjänster & Service” och valde sedan ”Fartygsoperationer” där han hittade det han sökte, dock med kommentaren att ”Fartygsoperationer” var ett vilseledande namn.

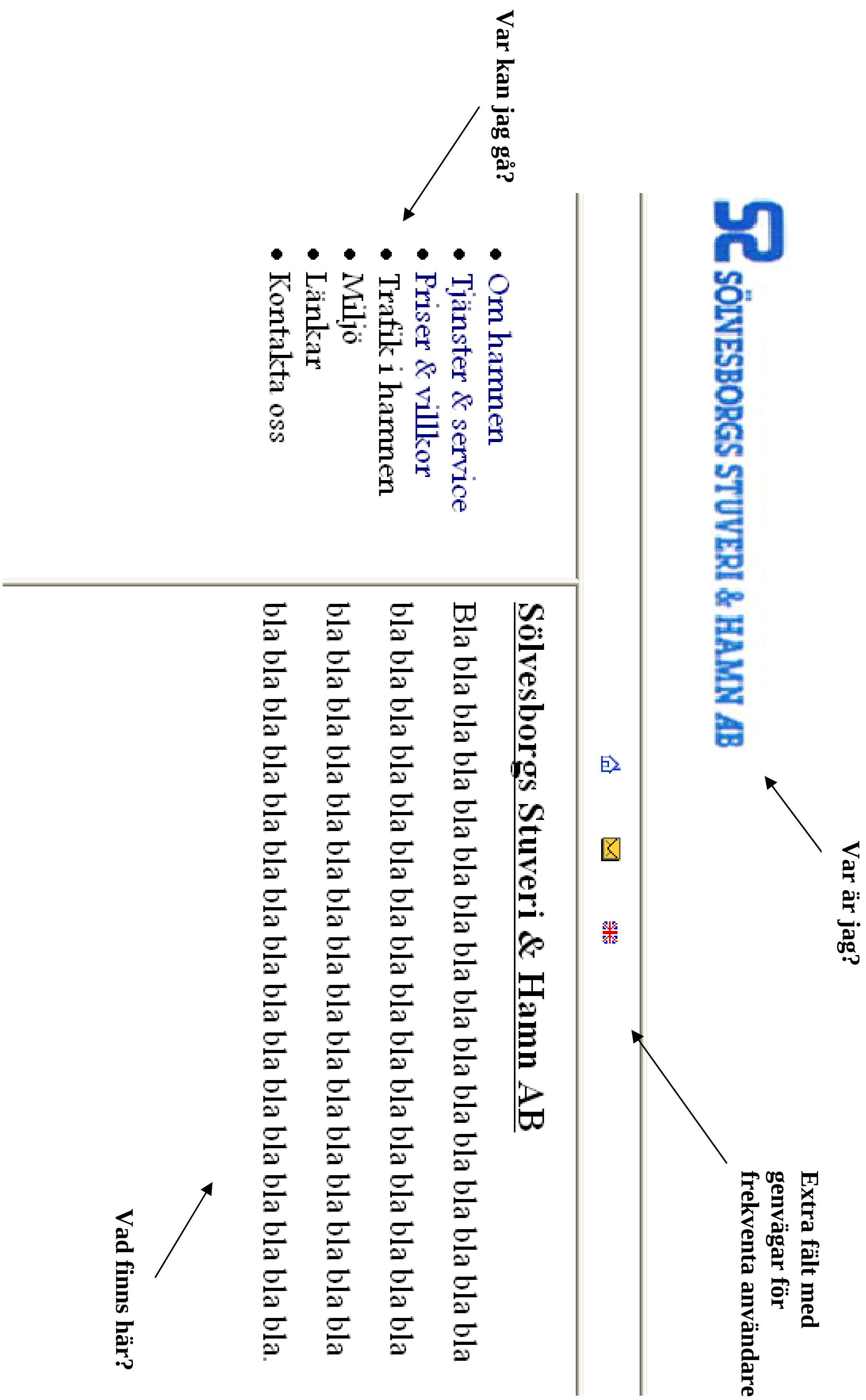
2: Personen gick in under rubriken ”Om Hamnen” och sedan in under ”Organisation”, där han hittade det rätta svaret.

3: Personen gick in under ”Tjänster & Service” och sen in under ”Resurser” och hittade det han sökte.

4: Personen gick med viss tvekan in under ”Priser & Villkor” då han även tyckte svaret kunde ligga under ”Om Hamnen”, men eftersom han tyckte det var ett villkor så testade han ”Priser & Villkor” där han mycket riktigt hittade svaret.

Övriga kommentarer
Övriga kommentarer gällande meny systemet var att rubrikerna under ”Om Hamnen” skulle vara i en annan ordning eftersom den mest relevanta rubriken ska vara överst och så vidare. Ordningen personerna tyckte rubrikerna skulle komma i var följande; ”Hamnfakta”, ”Organisation”, ”Historik”, ”Idag”, ”Framtid”, ”Säkerhet”, ”Bildgalleri”. En del kommenterade också att rubrikerna endast ska ha stor bokstav i början på första ordet i rubriken inte i början på varje nytt ord. De tyckte även att ”Organisationsshema” var en onödig länk. Information om vilken ställning varje anställd på SSHAB har ska istället synas på ”Kontakta oss” sidan. Önskemål om en länk kallad ”Utbyggnader” som skulle ligga under ”Framtid” gavs.

Appendix H – The design structure for the applications



Appendix I – The design of the homepage

Appendix J – The design of the logistic application

Appendix K – Questions and answers to the user test

Utvärdering av hemsidan & logistik applikationen

Kön Kvinna Man

Åldersgrupp -19 20-29 30-39 40-49 50-59 60-

1. Vad är ditt första intryck av applikationerna?

Trist 1 2 3 4 5 Mycket bra

Kommentar:_____
_____

2. Är utformningen av applikationerna bra, dvs. är de lätt orienterade, lätt förståliga, lätt navigerade och stödjer de ditt behov?

Ja Nej

Kommentar:_____
_____

3. Uppfattade du applikationerna som svåra att lära?

Ja Nej

Kommentar:_____
_____

4. Uppfattade du applikationerna som osäkra att interagera med, dvs. hamnade du på något ställe i en icke önskvärd situation?

Ja Nej

Kommentar:_____
_____

5. Är menyerna tydligt strukturerade?

Ja Nej

Kommentar:_____
_____

6. Är bilderna i informations delen av applikationerna bra?

Ja Nej

Kommentar:_____

7. Är ikonerna tydliga och bra placerade?

Ja Nej

Kommentar:_____

8. Är typsnitt och storleken på texten bra?

Ja Nej

Kommentar:_____

9. Ges tydlig nog feedback?

Ja Nej

Kommentar:_____

10. Kommer applikationerna att underlätta ditt dagliga arbete?

Inte alls 1 2 3 4 5 Väldigt mycket

Kommentar:_____
_____

11. Sätt ett kryss i den eller de rutan/orna som du tycker passar bra in på applikationerna:

☐ trevlig ☐ estetiskt tillfredställande ☐ hjälpsam
☐ emotionellt tillfredsställande ☐ motiverande ☐ givande

12. Övriga kommentarer:

_____
_____
_____
_____

Tack for din medverkan!

Svar:

Person 1 (Man, 20-29)
1. 5. Dock är ISO-märket på hemsidan ett gammalt märke som inte används mer. Ha istället kvalité som en länk i menysystemet, tex. under ”Om hamnen”.
2. Ja, bra och välstrukturerade designer.
3. Nej.
4. Nej.
5. Ja, jag är framför allt mycket imponerad av menyn på hemsidan - väldigt klar och tydlig.
6. Ja.
7. Ja.
8. Ja.
9. Ja.
10. 5.
11. Trevlig, emotionellt tillfredsställande, estetiskt tillfredställande, motiverande, hjälpsam, givande.
12. Snygg design!

Person 2 (Man, 30-39)
1. 4.
2. Ja.
3. Nej.
4. Nej.
5. Ja, bra med mindre font ju djupare man går i navigeringen.
6. Ja.
7. Ja.
8. Ja.
9. Ja.
10. 5.
11. Trevlig, emotionellt tillfredsställande, estetiskt tillfredställande, motiverande, hjälpsam, givande.
12. Tycker det är snyggt att hemsidan är centrerad, dock måste hänsyn tas till de som har en lägre upplösning på skärmen. Tycker inte bakgrunden ska ta så stor plats i detta fall och engelska flaggan måste synas med en gång så att inte de internationella kunderna måste skrolla för att hitta den.

Person 3 (Kvinna, 50-59)
1. 5.
2. Ja, lätt att hitta det man söker.
3. Nej.
4. Nej.
5. Ja.
6. Ja.
7. Ja. Tycker det är bra med text under så man aldrig behöver tveka vad de betyder.
8. Ja.
9. Ja.
10. 4.
11. Trevlig, emotionellt tillfredsställande, estetiskt tillfredställande, motiverande, hjälpsam, givande.
12. Tycker designerna av applikationerna är bra och tydliga.

Person 4 (Man, 50-59)
1. 5.
2. Ja.
3. Nej.
4. Nej.
5. Ja.
6. Ja. De ger en bra bild av hamnen.
7. Ja.
8. Ja.
9. Ja.
10. 4.
11. Trevlig, emotionellt tillfredsställande, estetiskt tillfredställande, motiverande, hjälpsam, givande.
12. -

Person 5 (Kvinna, 40-49)
1. 4.
2. Ja, där finns inte en massa onödig information.
3. Nej.
4. Nej.
5. Ja.
6. Ja.
7. Ja.
8. Ja.
9. Ja.
10. 4.
11. Trevlig, emotionellt tillfredsställande, estetiskt tillfredställande, motiverande, hjälpsam, givande.
12. -

Appendix L – Questions and results to the last user test

Frågor:

- **Är det något du skulle vilja ändra på när det gäller utformningen av applikationerna?**
- **Är et några funktioner eller tjänster som du skulle vilja lägga till eller ta bort på dem?**
- **Några sista önskemål vad gäller informationen på dem?**
- **Övriga kommentarer?**

Resultat:

Inga av användarna hade något ytterligare att komma med. De tyckte designen av applikationerna var mycket bra och de funktioner och tjänster de behövde fanns med.