Usability Engineering

Course No.: 188.314
Course Instructor: Mag. (FH) Bettina Thurnher
Quality Software Engineering (QSE) Research Group
TU Wien, Institute of Software Technology and Interactive Systems (IFS)

Last Changed: 20.09.2004
# Table of Contents

1  **Definition & Motivation** .................................................................................. 1
   1.1  What is Usability? ................................................................................. 1
   1.2  Usability Engineering and User Interface Design................................. 3
   1.3  Why Usability Engineering? ................................................................. 3
   1.4  The Benefit of Usability Engineering ................................................... 4
   1.5  Gestalt Gesetze ...................................................................................... 5
       1.5.1  Law of Figure and Ground.......................................................... 5
       1.5.2  Law of Proximity .......................................................................... 6
       1.5.3  Law of Similarity .......................................................................... 6
       1.5.4  Law of Closure ............................................................................. 6
       1.5.5  Law of Experience ........................................................................ 6
       1.5.6  Law of Good Continuation ......................................................... 7
       1.5.7  Law of good form ......................................................................... 7
       1.5.8  Illusions ........................................................................................ 8
   1.6  Cultural Aspects ..................................................................................... 8
       1.6.1  Names .......................................................................................... 8
       1.6.2  Addresses .................................................................................... 8
       1.6.3  Smilies ........................................................................................ 9
       1.6.4  Colors ........................................................................................ 9
       1.6.5  Websites ....................................................................................... 9
2  **The Usability Process** ............................................................................... 10
   2.1  Software Engineering Lifecycle.......................................................... 10
   2.2  Usability Engineering Lifecycle ......................................................... 10
3  **Usability Methods** ............................................................................... 13
   3.1  User Profiling .................................................................................... 13
3.11.2 Measurement of Interface Efficiency

3.11.3 Fitt’s Law and Hick’s Law

4 Usability and Process Models

4.1 What is a Process Model

4.2 The Problem with Business Process Models

4.3 A user-friendly Process Model

5 Mobile Usability Engineering

5.1 History of Mobile Applications

5.2 What is Mobile Usability?

5.3 Mobile Usability Testing

5.3.1 Technical Problems

5.3.2 Application vs. End Device

5.3.3 Scenarios: Simulation of Context

5.3.4 Visualization

5.4 Mobile Usability Prototypes

5.4.1 Paper Prototypes

5.4.2 Simulators

5.4.3 Functional Prototypes

5.4.4 Card Sorting

5.4.5 Performing the Test

6 Web Usability

6.1 What makes the Web different?

6.2 Guidelines

6.2.1 Design guidelines

6.2.2 Web Accessibility

7 References
Table of Figures

Figures

Figure 1-1: Traditional vs. human-centred development, [Gull01] ....................... 2
Figure 1-2: Negative and positive Images, [Jako_1] ............................................. 5
Figure 1-3: Groups of lines, [Webl03] ................................................................. 6
Figure 1-4: Lines, [Davi96] ................................................................................... 6
Figure 1-5: Interrupted lines as forming closed figures, [Davi96] ....................... 6
Figure 1-6: Completion of missing parts through experience, [Webl03] ............ 7
Figure 1-7: I’d better add ‘easy to use’ to the list, [Dilb01] ................................. 7
Figure 1-8: This is a “B”, [Boer00] ....................................................................... 7
Figure 1-9: Illusions, [Webl03] ............................................................................. 8
Figure 1-10: Changing size, [Webl03] ................................................................. 8
Figure 2-1: The usability engineering lifecycle, [Mayh04] ............................... 12
Figure 2-2: Usability Engineering Lifecycle, [Mann01] ..................................... 12
Figure 3-1: Different levels of realism, [Farm04] .............................................. 16
Figure 4-1: A user-friendly process model, [Thur03] ........................................ 31
Figure 5-1: Using cameras, [Litt04] ................................................................. 35
Figure 6-1: The Web Usability Checklist, [Andr04] ........................................ 39

Tables

Table 1-1: A model of the attributes of system acceptability, [Niel93] ............... 1
Table 1-2: Western and Japanese smilies, [Wiki04a] ......................................... 9
Table 1-3: Male and female Japanese smilies, [Wiki04a] .................................. 9
Table 3-1: Interactive Prototype made in Shockwave, [Edwa99] ...................... 17
Table 3-2: Interface timings, [Rask00] ............................................................... 29
1 Definition & Motivation

1.1 What is Usability?

The ISO 9241-11: Guidance on Usability [ISO9241] defines:

“Usability: 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.” [UsNe03]

For instance:

- Appropriate for a purpose
- Comprehensible, usable, (learnable), …
- Ergonomic, high-performance, ...
- Reliable, robust, …

“Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.” [Niel03]

![Table 1-1: A model of the attributes of system acceptability, [Niel93]](image)
Usability has five quality components [Niel03]:

- **Learnability**: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency**: Once users have learned the design, how quickly can they perform tasks?
- **Memorability**: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- **Errors**: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**: How pleasant is it to use the design?

![Figure 1-1: Traditional software development vs. human-centred development, [Gull01]](image)

According to Nielsen in [Niel03] there are also many other important quality attributes such as e.g. utility. In his point of view, there is no sense in having great features, if they are too difficult to use.

"**Usability is a measurable characteristic of a product user interface that is present to a greater or lesser degree. One broad dimension of usability is how easy to learn the user interface is for novice and casual users. Another is how easy to use (efficient, flexible, powerful) the user interface is for frequent and**"
proficient users, after they have mastered the initial learning of the interface”
[Mayh99]

1.2 Usability Engineering and User Interface Design

“Usability Engineering is a discipline that provides structured methods for achieving usability in user interface design during product development.”
[Mayh99]

User Interface Design describes any kind of interface design of an interactive system and also the information design of non-interactive systems (e.g. subway map). Thus User Interface Design is a subset of Usability Engineering.

1.3 Why Usability Engineering?

In the past years the importance and availability of computers have changed radical. Computers are not expert only systems, but they have a deep impact of every person’s daily life. Under these circumstances it is very important, that computers, software and interactive systems as a whole are simple to use and to learn. Software is becoming more complex; every new version of a software product has even more complex features. At the same time, Software is becoming more interactive, there has been a shift over the last decade from line and mask oriented software to graphical user interfaces. In addition there is a change of user groups, in the past typical users were computer experts using highly customized and custom-made software after having special software training. Today the target group of a software product is much greater and more heterogeneous; users are often not very experienced in using computers. Also there is often no time or motivation for special software training.
Five reasons by [Andr04] why there is a need for a structured process to increase usability and why it is not already done during the development process (e.g. by the developers themselves):

- People believe they understand the behavior of others based on their own experience
- This belief is only lost through prediction and measurement
- Experience changes one’s perception of the world
- Designers of system find it easy to use, naturally
- Often intuitions are wrong

1.4 The Benefit of Usability Engineering

Enhanced reliability and efficiency of the system will have economical effects by decreasing user support costs, and time-consuming training investments. Usability engineering will decrease technology development time and costs: late discovery of serious flaws to a system will render the necessary re-engineering time- and cost-intensive.

A proper separation of layers (e.g. the separation of the data layer, business logic and presentation layer) will make changes to the application easier. In web development the business logic and the presentation layer are often not clearly separated, thus making later changes (e.g. usability improvements) costly.

- Every $1 invested in user-centered design returns between $2 and $100 [Press92]
- Users will
  - Experience satisfaction instead of frustration
  - Achieve goals more effectively and efficiently
  - Not waste time and energy
  - Easily learn to handle the system
- Providers/producers/developers will benefit from
  - Reduced financial costs
  - Efficient design that adds value, not frills
1 - Definition & Motivation - Gestalt Gesetze

- Fewer revisions
- Reduction of support costs
- Increased productivity
- Increased accessibility to maximize the potential audience
- Increase in use
- Happy and loyal customers
- Reduced development times
- Avoidance of unnecessary features

1.5 Gestalt Gesetze

The following “Gestalt Gesetze” which are part of the “gestalt psychology” try to give a better understanding how the human brain processes information. This knowledge facilitates the design of interactive systems with good usability.

1.5.1 Law of Figure and Ground

Primarily the human brain percept figures instead of ground. Depending on the choice of colors forms can be interpreted ether as figures or ground.

![Figure 1-2: Negative and positive Images, [Jako_1]](image-url)
1.5.2 Law of Proximity

Close objects are perceived as groups

![Figure 1-3: Groups of lines, [Web103]](image)

1.5.3 Law of Similarity

Similar objects are perceived as groups. Differences have higher information content, thus the perception costs are higher.

![Figure 1-4: Lines, [Davi96]](image)

1.5.4 Law of Closure

Missing parts of an element are supplemented according to experiences of possible figures of perception.

![Figure 1-5: Interrupted lines as forming closed figures, [Davi96]](image)

1.5.5 Law of Experience

This law explains why we read over spelling mistake and typos. The human brain reads automatically the correct word, although it is not spelled correct. This means
that we don’t analyze every perception anew, but we anticipate what seems probably according to our experiences.

**Figure 1-6: Completion of missing parts through experience, [Webl03]**

### 1.5.6 Law of Good Continuation

Objects which are in a temporal context can be interpreted as motion (e.g. comic strips, movies)

![Dilbert Comic Strip](image)

**Figure 1-7: I'd better add ‘easy to use’ to the list, [Dilb01]**

### 1.5.7 Law of good form

This law means that a row of points or objects, which add up to a line when connected, is perceived as belonging together.

![Line Drawing](image)

**Figure 1-8: This is a “B”, [Boer00]**
1.5.8 Illusions

Bright objects on dark background appear to be larger.

Figure 1-9: Illusions, [Webl03]

Objects change their subjective size depending on surrounding objects.

Figure 1-10: Changing size, [Webl03]

1.6 Cultural Aspects

In the design of a system it is also important to consider cultural aspects and differences. The following examples show how these differences could be. It seems obvious that a good and usable design has to be considerate of cultural aspects.

1.6.1 Names

In China the name order is surname before forename while in western countries it is vice versa. In other countries, like Oman there are no last names at all.

1.6.2 Addresses

In Russia it is common practice to write first of all the land and in the end the name of the recipient.
1.6.3 Smilies

Western smilies are very different to Japanese ones. While western smilies are lying on the side, Japanese smilies are in the “right position”.

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Western smiley</th>
<th>Japanese smiley</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smile</td>
<td>: - )</td>
<td>^_.^</td>
</tr>
<tr>
<td>Wink</td>
<td>; - )</td>
<td>`_.^</td>
</tr>
<tr>
<td>Crying</td>
<td>' - (</td>
<td>;_;;</td>
</tr>
</tbody>
</table>

Table 1-2: Western and Japanese smilies, [Wiki04a]

Japanese smilies often use a period as a mouth to make it look cuter. This can indicate that it is representing a female.

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smile</td>
<td>^_.^</td>
<td>^_.^</td>
</tr>
<tr>
<td>Wink</td>
<td>`_.^</td>
<td>`_.^</td>
</tr>
<tr>
<td>Crying</td>
<td>;_;;</td>
<td>;_;;</td>
</tr>
</tbody>
</table>

Table 1-3: Male and female Japanese smilies, [Wiki04a]

1.6.4 Colors

In western countries green is the color for hope. In Indonesia green is the color of danger. In India white is the color of sadness and sorrow, in western countries it is black.

1.6.5 Websites

For Chinese websites it is important to have commercial banners, because they are considered as good product information. Websites without commercial banners are less successful.
2 The Usability Process

2.1 Software Engineering Lifecycle

Software engineering describes the process of application developing and although there are different kinds of methods, processes and ideas, most of the methods include these steps:

1. Requirements Analysis, what to design and why, and what goals should be achieved
2. Development of a Concept; how to get to a product
3. Implementation of the Concept
4. Testing and Debugging
5. Delivery and Installation
6. Maintenance, i.e. evaluation of the current installation.

This will lead to a product or an updated version of the product, since most of this software engineering processes are iterative processes\(^1\), so when one iteration is finished a new repetition of the software engineering life cycle will start.

2.2 Usability Engineering Lifecycle

Similar to the software engineering lifecycle, the Usability Engineering Life Cycle goes hand in hand with the software being developed and as like as there are many different software engineering methods and process models, there are also several usability engineering life cycles. The often have the following phases in common:

\(^1\) Of course there are also non-iterative models e.g. waterfall model.
1. **Requirements Analysis**
   - Who is the end user, who will use the product? **User profiles** are conceived.
   - **The Task analysis** reveals task requirements and the current workflow as well as specific conditions of use.
   - **Platform capabilities and constraints** are recognized at this phase.

2. **Conceptual Model Design** includes reengineering the workflow and writing down the application structure.

3. **Testing and Debugging** in Human Computer Interaction means designing rapid prototypes and users test them.

4. **Implementation** of the "final" system, delivery and installation.

5. **Evaluation** and refinement of the requirements lead to an update of the product, repeating the usability engineering life cycle.

*Mayhew structured the "Usability Engineering Lifecycle" [Mayh99] into three main sections, "Requirements Analysis", "Design/Testing/Development", and "Installation".*

The workflow is rather complex, within each section there are a number of different tasks to accomplish. These tasks cover structured usability requirements analysis task, goal settings, the development of a user interface to match these goals and finally evaluation tasks. Many of the methods used in were explained in 3 - Usability Methods, p. 13.

Phase One, the requirements analysis includes the creation of a user profile (see 3.1 - User Profiling, p. 13), a contextual task analysis (see 3.4 - Task Analysis, p.15), the specification of *qualitative* and *quantitative* goals and the platform capabilities and constrains. These four tasks will lead to the “Style Guide”.

Phase two is divided into three “levels”: design, testing and development. It starts in level 1 – design – with “work reengineering”, a conceptual model design, conceptual model mock-ups (prototypes) and an iterative conceptual model evaluation with usability testing, to improve the model. Level 2 is concerned with
the setting of standards, and this screen design standards will be tested with prototypes. Also an iterative evaluation will ensure the quality. In level 3 the detailed user interface design will be developed and iteratively evaluated. Finally in phase three user feedbacks is gathered for enhancements.

Figure 2-1: The usability engineering lifecycle, [Mayh04]

A less complex model of a usability engineering lifecycle is the model of Manhartsberger, The evaluation of the user interface starts in a very early stage of the development process (paper prototypes…). This model is also incremental.

Figure 2-2: Usability Engineering Lifecycle, [Mann01]
3 Usability Methods

3.1 User Profiling

User profiling is described as “…a process of gathering all known information about the audience of a specific product, and then breaking them into specific ‘profiles’.” [Usab_1]

For example a site selling books and writing tools may attract three main types of visitor - resellers, students, and general members of the public.

A document is then created for each of these groups, and is built up to including information such as:

- Technology used by these users
- Population characteristics (age, gender, socio-economic breakdown, etc.)
- Relevant experience (in the industry, with the technology, etc.)

As well as gathering information on each profile as a sub-section of the audience, each profile document also includes a “persona”.

3.2 User Persona

A persona is a made-up person who represents a centre-line view of a particular user profile. It is a description of a specific person who is a target user of a system being designed, providing demographic information, needs, preferences, biographical information, and a photo or illustration. Typically, multiple personas are developed in the early stages of design that represent the spectrum of the target audience. Personas are one piece of a "scenario"; the other piece is a description of how this person would typically interact with the system being designed.
The point of developing personas is to avoid the trap of designing for the "average" user that does not actually exist, and instead to make sure that the system will work for somebody specific rather than no one in particular [Diam_1].

For example a site that sells books over the Internet may have a user profile of a “student”. Some of the characteristics of the student who uses the site (together with many other users) might be:

- Age range: 18-30, average age 23.
- 48% male, 52% female.
- Average income per month in EUR:
  - Small, less than 300 - 62%.
  - Medium, between 300 and 1000 - 31%.
  - Large, more than 1000 - 7%.

Using that information and more, a persona can be created that represents this profile:

“Heinzi is a 24 year old student with a monthly income of 500 EUR. Heinzi is interested in books about computer science in general, but his favorite topic is Linux.”

Personas turn abstract data into “real people”, and help you to focus on your customers without skewing your view.

3.3 User Scenarios

“A scenario is an encapsulated description of an individual user, using a specific set of computer facilities to achieve a specific outcome under specified circumstances over a certain time interval (this in contrast to simple static collection of screens and menus: The scenario explicitly includes a time dimension of what happens when).”[Nie93]

With the data collected from the user profiling and the so created persona it is possible to play different scenarios. In these scenarios a persona uses the software to reach a specific goal. In User testing, when the tester is a real person (see 3.10.3
Testers, p. 25) it is also a possible alternative to develop a scenario together with the tester. This is of course very time consuming but is a trade off between time and testing quality worth considering.

**Daily use scenario**

Tasks the users are executing most of the time are called Daily Tasks. Thus in software development the main focus is targeted on these scenarios. Daily scenarios should be straightforward to access and quick to handle. Example: Customer wants to order a book about the Apache server.

**Necessary use scenario**

These are important actions, which are somewhat necessary to complete the user’s tasks but are not used most of the time. Example: Customer changes his credit card information, because his old card expired.

**Edge case scenario**

Tasks executed infrequently are named edge case scenarios. Although these tasks are seldom they should still be considered in software development and usability testing. Example: Customer wants to order a book which is not listed in the catalog.

### 3.4 Task Analysis

The process of identifying how people think about tasks and how they complete them called task analysis. The goal is make sure that the interface designed matches the users’ mental model of that task closely, so that the users can quickly and easily see what to do at every point. Example: when the book store wants to sell goods in an online store, then task analysis would be used to identify the process that people go through in deciding to buy. It would be used to identify what information they needed at each point, what their concerns were at every step, and what options they used.
3.5 Prototyping

Prototypes allow testing in early development stages on a low price and effort and also to test different concepts. Prototypes cut down either on the number of features or the depth of functionality of features to perform usability evaluation as early as possible in the design cycle. Prototypes with different levels of realism, interactivity and production effort can be used according to the current test.

![Diagram showing different levels of realism](image)

**Figure 3-1: Different levels of realism, [Farm04]**

3.5.1 Vertical prototype

“Cutting down on the number of features is called vertical prototyping since the result is a narrow system that does include in-depth functionality, but only for a few selected features”. [Niel93]

So a vertical prototype demonstrates the exact functionality of a product but for only a small section of the entire product. For example, a vertical prototype of a word processor might demonstrate all of the spell-checking functions, but none of the formatting or text-entry functions. All of the functions in a vertical prototype mimic their real counterparts as much as possible.

3.5.2 Horizontal prototype

“Reducing the level of functionality is called horizontal prototyping since the result is a surface layer that includes the entire user interface to a full-featured system but with no underlying functionality. “ [Niel93]
Horizontal prototypes demonstrate a broad spectrum of the product’s features, but without extensive functionality behind each function. Horizontal prototypes are often used for user preference testing of user interfaces, when the actual working functions have not been implemented yet. Such prototypes allow evaluation of the interface design, feature placement and accessibility and the like, without requiring anything to actually work.

### 3.5.3 Scenario prototype

A scenario is a mixture between vertical and horizontal prototypes. In a scenario a specific user task will be demonstrated as real as possible.

### 3.5.4 Types of prototypes

Depending on the current phase of a development process different prototype techniques may be used. Starting with verbal prototyping or paper mock-ups (sketches…) in early stages of the process over to fake data or simple algorithms and so on…

![Interactive Prototype made in Shockwave, [Edwa99]](image)

Table 3-1: Interactive Prototype made in Shockwave, [Edwa99]

A “Wizard of Oz” prototype is a prototype that only works by having someone behind-the-scenes who is pulling the levers and flipping the switches. The wizard of oz technique has a user interacting with an interface without knowing that the responses are being generated by a human, not a computer. This allows testing of some difficult interface concepts before a system is fully working.
3.6 Competitive Analysis

Competitive analysis is a set of methods for getting ideas from the competition, whether through reverse-engineering their user interface design rationale, determining common user interface conventions, or finding usability problems that can be fixed. Results of a competitive analysis can be used in marketing, by demonstrating the superior usability of your product, or for design, by getting ideas for ways to improve your product.

3.7 Style Guide

A style guide is a reference which defines the look-and-feel of a user interface by clearly setting the standards and conventions of that specific user interface. Style guides usually include the principles that guide the design of the interface, graphic layout grids, exact size and spacing of elements in the interface, fonts, colors, interactive behavior, standard text messages (such as error messages), and labeling standards. Excellent style guides are available for most major operating systems. Specific style guides are often created for individual applications or websites to establish consistency in organization and visual standards, and to assist the user interface designer in communicating the design both to developers and to other designers who may be working on the same project in the future.

3.8 Design Workshops

When making a design workshop, the developers and the customer come together in order to increase the usability. The development team show design prototypes and scenarios and discuss them with the customer. These workshops provide valuable feedback, which is important especially in early stages of development.
3.9 Usability Inspection

In usability inspection experts use several techniques for evaluating a user interface by examining and critiquing it. The critique would normally be based on experience, psychological principles, a set of previously-defined guidelines or using heuristics and judgment.

3.9.1 Heuristic Evaluation

Nielsen defined ten general principles for user interface design [Niel_1]. He called them "heuristics" because they are “...more in the nature of rules of thumb than specific usability guidelines”.

3.9.1.1 Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. According to [Niel_2] the response time of a system should be of course as fast as possible. Is the response time 0.1 second and below the user feels that the system is reacting instantaneously, meaning in this case no special feedback is necessary, except to display the result. Up to one second it is about the limit for the user’s flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data. Everything longer than one second requires instant feedback. Examples for appropriate user feedback are dialog boxes, status messages, changing animated mouse pointers, progress bars...Also for longer delays the user wants to perform other tasks while waiting for the computer to finish (not necessarily on the same machine if the operation consumes all the CPU time) so the user wants to have feedback indicating when the computer expects to be done. Example: An average user will wait a maximum of 3 seconds for a website to load. If the website takes longer to load, the user will surf on and in about 80% not come back.
3.9.1.2 Match between system and the real world

The system should speak the user’s language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. Code developers often speak their own language (also known as techno-babble) which makes it hard for others to follow. So the system language should reflect the target user’s language.

3.9.1.3 User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo and clear navigation, so that the users always have the feeling to “be in command” of a situation.

3.9.1.4 Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. For most platforms e.g. operation systems like Microsoft Windows there are extensive guidelines and style guides of application and user interface design, so that users are able to switch the application without the need to learn new standard shortcuts or a different file menu dialog.

3.9.1.5 Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. To minimize errors advanced software engineering methods should be used when creating an application. All newer software engineering methods implement test strategies, for example extreme programming.

---

2 Windows XP - Guidelines for Applications
http://www.microsoft.com/whdc/hwdev/windowsxp/downloads/default.mspx
3.9.1.6 Recognition rather than recall
Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

3.9.1.7 Flexibility and efficiency of use
Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Accelerators can be for example keyboard shortcuts which allow more experienced users to interact much faster with the system.

3.9.1.8 Aesthetic and minimalist design
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. Bad examples of “too much design” can often be seen on websites with “Flash”. Because of great animation and a lot of shiny colors, the sites become hardly usable.

3.9.1.9 Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

3.9.1.10 Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation especially on more complex systems. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.

3.9.2 Guideline Reviews
Guideline reviews are inspections where an interface is checked for conformance with a comprehensive list of usability guidelines. However, since guideline documents contain a few hundred guidelines, guideline reviews require a high degree of expertise and are fairly rare in practice. The method can be considered
as somewhat of a cross between heuristic evaluation and standards. Sometimes the companies’ style guide does also include a usability section which can be used for reviews.

### 3.9.3 Pluralistic Walkthroughs

These walkthroughs are meetings where users, developers and human factors experts step through a scenario, discussing usability issues associated with dialog elements involved in the scenario steps.

### 3.9.4 Consistency Inspections

Consistency inspections have designers representing multiple projects inspect an interface to see whether it does things in a way that is consistent with their own design. Thus, consistency inspections are aimed at evaluating consistency across the family of products that has been evaluated by an inspection team.

### 3.9.5 Standards Inspection

Standard inspections have an expert on some interface standard inspect the interface for compliance. Thus, standards are aimed at increasing the degree to which a given interface is in the range of other systems on the market that follow the same standard.

For example, software products designed for the Windows OS should have common elements, such as the same file dialogs, a Help menu, the Help menu can be called with the F1 key etc. Or, products designed to be marketed in a particular country may have to conform to that country’s ergonomic standards. Many monitors or keyboards are restricted from some uses in certain European countries due to occupational safety and health standards in those countries.

### 3.9.6 Cognitive Walkthroughs

Cognitive walkthroughs use a more explicitly detailed procedure to simulate a user’s problem-solving process at each step in the human-computer dialog, checking to see if the simulated user’s goals and memory for actions can be
assumed to lead to the next correct action. Note: This is only a “simulated user”, there is no real user involved.

### 3.9.7 Formal Usability Inspections

Formal usability inspections are intended to be very similar to various code inspection techniques known from software engineering methods. Various participants have well defined responsibilities and roles. A moderator is appointed to manage the focused inspections and the inspection meeting; a design owner is responsible for design and redesign; the inspectors are responsible for finding problems; and a scribe records all defects and issues identified during the meeting.

### 3.9.8 Feature Inspections

Feature inspections analyze only the feature set of a product, whether the function designed meets the needs of intended users and whether they match the expected user scenarios. For example, a common user scenario for the use of an online shop is to buy a product. The features that would be used include putting something into the basket, check-out, give credit card and shipping information, review the order and finally approve the order. Each set of features used to produce the required output is being analyzed for its availability, understandability, and general usefulness.

### 3.10 Usability Testing

#### 3.10.1 Purpose of Testing

“Usability test” is the empirical testing of interface design with representative users. It is a process which should discover if given usability targets are fulfilled. It encompasses a range of methods that examine how users in the target audience interact with a system. In a typical approach, a user performs a variety of tasks with the application while an observer records notes on how successful the user is

---

3 For more information see [Zuse03]
in performing each task: how fast users go, what mistakes they make, where they are confused, what solution paths they take, how many learning trials are involved, and so forth.

"Usability testing is a means for measuring how well people can actually use something (such as a web page, a computer interface, a document, or a device) for its intended purpose. If users, or test subjects, have difficulty understanding instructions, manipulating parts, or interpreting feedback, then the developers must go back to the drawing board, improve the design, and test it again. During usability testing, developers are not expected to explain their product to the user, or argue about its merits." [Wiki04c]

**Formative Evaluation**
- Helps improve interface design
- Learn why something went wrong, not just that it went wrong
- Collect process data – qualitative observations of what happened and why

**Summative Evaluation**
- Assesses overall quality of interface
- Compare alternative designs, test definite performance requirements
- Collect bottom line date – quantitative measurements of performance (success, time to achieve, errors made)

### 3.10.2 Difference to Usability Inspection

In usability inspection usability *experts* analyze if the usability guidelines are implemented correctly (see also 3.9 Usability Inspection, page 19). In contrast usability testing concentrates on “*real*” test *users*. Together with this participant it should be evaluated, if the tested application is designed in the sense of usability.

Problems of testers using an application can indicate design and usability problems. Observing test users helps improving applications.
3.10.3 Testers

Testers should be selected adequate to the target group of the tested product. There is no sense in evaluating the design of controls and instrument of a car's cockpit, when the tester has no driving license and thus no personal interest and experience in with cars. It is recommended dividing testers into various groups due to age, educational and cultural background and – if the product is somehow computer related – computer experience. The testers’ opinion about a product should always be regarded in the context of his personal background and experiences.

To give test users more motivation and the possibility empathize in the situation of a person using the tested program, it is recommend developing user scenarios (see 3.3 User Scenarios, page 14). The user should know what he wants to do with the application and should know the desired outcome. The more real a user scenario is, the better a real-life situation can be simulated.

3.10.4 Test methods

3.10.4.1 Thinking Aloud Method

The Thinking Aloud method is a popular (and cheap) technique used during the whole usability process. During the course of a test, where the user is performing a task as part of a scenario, the user vocalizes his or her thoughts, feelings, and opinions while interacting with the product. One of the main benefits of the thinking aloud protocol is a better understanding of the user’s mental model and interaction with the product. Other benefits includes the understanding of the terminology the user uses to express an idea or function, which could lead to a better product design or at least better documentation.

3.10.4.2 Observation

Usability engineers go to representative user’s workplace and observe them working with the product, to understand how the users are using the system to accomplish their tasks and what kind of mental model the users have about the system. After the observation there should be the following results: how long takes it to fulfill a task, how many errors were made…

20.09.2004
3.10.4.3 Query Techniques, Interviews and Focus groups

Query techniques involve questioning the user directly about their experience of using the system under evaluation. This may be done, e.g. face to face as an interview, or in writing as a questionnaire.

In interviews and focus groups the users are queried about their experiences and preferences with a specific product. Opposed to surveys and questionnaires, the developers might be present to interact and facilitate discussion about various issues raised by your questions. With multiple users present, as with focus groups, the interaction among the users may raise additional issues, or identify common problems that many people experience.

3.10.4.4 Eye tracking

Eye tracking allows testers to identify what participants look at during the course of a usability test. Several different technologies are used, including skin electrodes, marked contact lenses, image processing cameras, and reflector trackers. The last type is probably the most effective, as it does not require physical contact with the user’s eye/eye socket. Instead, a beam of light is projected onto the eye; a sophisticated camera picks up the difference between the pupil reflection and known reference points to determine what the user is looking at.

3.10.4.5 Paper-and-Pencil Test

In this test a group or individual evaluation of a design that is not yet prototyped or otherwise developed is being tested to gain user feedback on an emerging design at the very earliest stages.

3.10.5 Usability Testing Process

Usability testing is a process in six steps to audit if an application fulfills usability guidelines.

3.10.5.1 Preparation

In this phase the test goals and the tasks which have to be accomplished to reach these goals have to be defined. Depending on the tasks and goals the proper test
method has to be chosen. Also an appropriate environment (noise, light …) should be simulated. It is recommended to create use cases. Use case makes it possible to understand how application should work under a given context.

### 3.10.5.2 Introduction

During the introduction the test is explained to the test user. The user should be informed which tasks he has to fulfill during the test and that he has to use the test material.

### 3.10.5.3 Test

During the test there should be as little interaction as possible between the test user and the interviewer. Any support given to the test user may influence the test result. However, sometimes it is necessary to intervene and remind the test user to follow the test demands (e.g. if the user stops talking during a Thinking Aloud method). It is also possible to help the user if he is stuck in a deadlock and does not know how to continue his task. These helpings have to be noted in the test report.

### 3.10.5.4 Debriefing

After the test there should be the possibility for the test user to give his opinion about the tested application. The impressions of the tester and suggestions for improvements can be very valuable and should be recorded. Furthermore, special situations which occurred during the test can be discussed – maybe why the user was confused while executing certain tasks.

### 3.10.5.5 Analysis

In the analysis the focus should be on the following points:

- **Time on Task** -- How long does it take users to complete a set of basic tasks? (For example, find something you want to buy, create a new user account, and order the item.)

- **Accuracy** -- How many mistakes did users make? (Can the user correct these errors, if given the proper feedback, or are the errors fatal?)

- **Recall** -- How much information does the user remember, after completing the assigned tasks?
• **Emotional Response** -- How does the user feel about the tasks completed? (Confident? Stressed? Would the user recommend this system to a friend?)

[Wiki04c]

### 3.10.5.6 Report

The report is the output of the usability test. The report should be a complete description of the test and test setting. It should cover all applied test methods, the test details and the results of the analysis. The report has to point out where the testers had usability problems with the application. Although a recommendation for redesign should be contained.

### 3.11 Quantitative Analyses

Most of the methods, heuristics and techniques in the preceding chapters gave qualitative results. Quantitative methods could cut discussion, about what might be better to a simple calculation. Another huge benefit is that “…understanding why the quantitative methods works guides us to understand important aspects of how humans interact with machines.” [Rask00]

In other words: while qualitative methods are helpful for analyzing and understanding interface design, quantitative methods are helpful in understanding human interactions with machines.

#### 3.11.1 GOMS Keystroke Model

The GOMS (Goals, Objects, methods, selection rules) modeling allows to predict how long it takes for an experienced worker to accomplish a specific task when using a given interface design. The simplest GOMS method is the keystroke model. Tasks are composed of elementary gestures. People perform tasks at different speeds, but typical times, rather than individual times, work well for comparative analysis. So the keystroke model defines (typical) interface timings.
K = 0.2 sec  Keying: The time it takes to tap a key on the keyboard

P = 1.1 sec  Pointing: The time it takes a user to point to a position on a display

H = 0.4 sec  Homing: The time it takes a user’s hand to move from the keyboard to the GID (Graphical Input Device) or from the GID to the keyboard

M = 1.35 sec  Mentally preparing: The time it takes a user to prepare mentally for the next step

R  Responding: The time a user must wait for a computer to respond to input

<table>
<thead>
<tr>
<th>Table 3-2: Interface timings, [Rask00]</th>
</tr>
</thead>
</table>

With this setting it is possible to do calculations; there is also a heuristic for placing the time for the mental preparation for the next step (the “Ms”) into the process. Example: \( H + K + K + M + H + P = 3.65 \text{sec} \)

### 3.11.2 Measurement of Interface Efficiency

“The information efficiency \( E \) of an interface is defined as the minimum amount of information necessary to complete a task divided by the amount of information that has to be supplied by the user” [Rask00]

\( E \) is at least 0 and at it is most 1. \( E \) can be 0 when the user has to do something totally unnecessary. [Rask00] describes techniques to calculate this efficiency by taking the amount of information into calculation. The information is measured in bits or characters.

### 3.11.3 Fitt’s Law and Hick’s Law

Fitt’s law returns the average time it takes a user to get the cursor to the button, while Hick’s law is concerned about the amount of alternative actions the user might have.
4 Usability and Process Models

4.1 What is a Process Model

“A process is a naturally occurring or designed sequence of operations or events, possibly taking up time, space, expertise or other resource, which produces some outcome. A process may be identified by the changes it creates in the properties of one or more objects under its influence.” [Wiki04d]

A business process is a process which wants to achieve a commercial result; a business process model typically defines the following elements:

- Specific inputs
- Specific outputs
- Resources consumed
- Activities that are performed in some order
- Events that drive the process.
- Customer oriented (internal or external)
- process is measurable

4.2 The Problem with Business Process Models

When developing new applications it can be very difficult to find a visualization which is clear and understandable for all participants. In software engineering e.g. the UML notation is widely used, but there will always be important participants on projects, who have not the insight in this kind of diagrams. This knowledge
gap appears especially high when developer and sales people work together on a software project.

4.3 A user-friendly Process Model

To avoid these problems an example for an iterative-incremental process design cycle could look like that:

![Figure 4-1: A user-friendly process model, [Thur03]](image)

In addition this process incorporates a controlling of the process workflow [Thurn03].
5 Mobile Usability Engineering

5.1 History of Mobile Applications

The history of mobile devices starts in the early fifties of the last century with the development of pagers for medical facilities [Wiki04b]. By calling special pager phone numbers the caller’s number was showed on the small pager display. Also in the fifties the first “mobile” phone was released, but mobile in a quite different sense than today. These mobiles had the size which allowed only the installation in cars. Also the coverage was not very well expended and calls had often been disrupted. In the late eighties and early nineties cellular phones become smaller and more comfortable and thus very widespread. In 2002 already 82% of the Austrian citizens owned a cellular phone [FMK_1]. Today mobile phones offer a large number of applications like taking photos, sending and receiving data, serving the internet or listen to music.

Since the nineties PDAs (Personal Digital Assistants) have been developed; computer handhelds which represent the functions of a full sized desktop computer in a reduced form. Beneath cellular phones, PDAs are the main field of mobile applications.

5.2 What is Mobile Usability?

To understand why mobile usability is an own sub area of usability, you have to understand the special circumstances of mobile applications. Using mobile application differs a lot to the use of not mobile applications. To visualize these differences think of two different scenarios:

On the one hand, a non mobile application user could be a person in front of a desktop computer in an office fulfilling certain tasks. The user has a quiet
environment and can fully concentrate on his work. Further it is possible to consult a book or co-workers, if the user needs some additional information. Although there is no time to waste for him, he has no explicit time pressure.

On the other hand, the mobile application user: Think of an insurance salesman, taking the subway to get from one customer to another. To spend this time productively he is using his mobile phone. The noise level is high, maybe he has no seat and has to stand and he is also under time pressure – he must not miss his station. When he gets out of subway he has to disrupt his task and to continue it later.

5.3 Mobile Usability Testing

Mobile usability testing is similar to standard usability testing. But as there are differences between mobile and non mobile usability there are some points in mobile usability testing which should be observed.

5.3.1 Technical Problems

During the planning of mobile usability test technical problems should be considered. They can influence the test itself and also the tester’s opinion. The reliability of the tested service should be guaranteed before starting the test. If the service gets unavailable during the test, maybe the test sequence can be rearranged. If it is likely that problems can occur, alternative proceedings should be planed. It is also very important, that the interviewer has not to take care of technical problems and can fully concentrate on the test person.

5.3.2 Application vs. End Device

During every test it is important to point out the test objective: the application, the end device or the service provider. This can be very difficult to distinguish for some testers.

It can also be the objective of a test, to clarify if the test person can distinguish between application and end device. If the layout of a tested application resembles
the user interface, the tester may not be able to distinguish the application from the end device.

When the end device is not the object of the test, the user should use a familiar device. If this is not possible, a warm-up phase should be given.

Here are four strategies for choosing the end devices:

1. Identify which devices 80% of your users will use to access your application. If this list is very short - 3 or fewer devices - plan on testing with all the devices on the list.

2. Test certain things, such as under which heading a particular page belongs, on paper without a device at all. Be cautious when testing labeling this way, since a reasonable label on one device may collide with the operating system labeling on another device.

3. Identify device families between which the user experience is the same or similar. You can assume that all Palm OS devices are largely the same, and all Pocket PC devices are largely the same. You can also identify families of phones that have the same characteristics, such as the Samsung 3500 and 8500. When you decide which of a device family to test with, choose the most limited - slowest, smallest display, fewest features, most difficult input - unless you can afford to ignore users of that device.

4. When doing iterative testing, use different devices for each test. Remember that you are looking for major problems. You are not trying to make a statistical statement about user behavior. You are instead saying "if some users in our test found a problem, then some users will find this problem."

[Litt04]

5.3.3 Scenarios: Simulation of Context

As mentioned before, the context of using a mobile device is important. It makes a difference using a device in a quiet labor situation where the user can fully
concentrate on his task and using a device on a public place with a high noise level. Observing the user’s behavior in a real scenario is quite difficult. It is also important that the scenarios match to the respective user. The better a user can relate to scenario the more he will be motivated during the test.

5.3.4 Visualization

The small size of mobile devices is not only a problem for the tester, who has to test it; it is also a problem for the interviewer to keep track on the user’s actions. Watching the user is one of the most important things in a usability test. The verbalization (think aloud) is a helpful method, but is not suffice and can only be used additional to observations. Also a looking over the user’s shoulder is inadvisable, because this would influence the user’s behavior. The best solution for this problem is, to film the test process by using a small camera, attached to the end device. Often a second camera filming the user’s body language is helpful to document his reactions.

![Figure 5-1: Using cameras, [Litt04]](image)

Alternative to film the end device display it is possible to use a computer simulations of the end device and record the user’s course of actions with a screen cam program. Although this is an inexpensive and simple method, it is quite inappropriate for testing mobile devices. The user is intended to hold the handheld in its hands like he would normally use it.

5.4 Mobile Usability Prototypes

A prototype is a functional limited representation of an application in development (see also the Chapter about Prototypes in 3.5 Prototyping, page 16).
Keep in mind that a prototype with a realistic appearance is expected to have realistic behavior and thus can disappoint the user’s expectations, especially with mobile devices. Often it is better to make prototypes look like ones.

### 5.4.1 Paper Prototypes

For early usability testing prototypes made of paper are the easiest and most inexpensive way for usability testing. Both the mobile device and the user interface can be simulated by paper prototypes. A separated paper prototype for the interface and mobile device allows simulating simple functions like scrolling.

*Appropriate to test:* wording, parallel testing of several concepts, iterative testing  
*Not appropriate to test:* handling, look and feel of application on end device, reliability, loading times, design  
*Advantages:* quick to make, cheap, changes to prototypes are easy to make, even between two tests of a session, reliable, low tech avoids disappointments

### 5.4.2 Simulators

Simulators can cover simple PowerPoint presentation with all required click paths as well as more sophisticated simulation tools simulation the mobile device on a desktop computer.

*Appropriate to test:* icons, wording, and workflow  
*Not appropriate to test:* look and feel of application on end device, loading time, compatibility  
*Advantages:* easy to record, testing on different (simulated) end devices, offline testing (no online service needed)

### 5.4.3 Functional Prototypes

Functional prototypes are very realistic: they can be tested on mobile device, but are mostly local limited and in simulated networks. Functional prototypes are costly to develop and can not be simple modified. When working with mobile internet applications it is also hard to keep services available.
Appropriate to test: specific end device testing, real interaction, design, and reliability, in parts also loading times

Not appropriate to test: everything which can be tested cheaper

Advantages: realistic testing scenarios with end devices

### 5.4.4 Card Sorting

Card sorting can help to evaluate how information should be arranged within a mobile application. Users have to cluster cards, according to the represented information, that should be arranged together, thus creating an on-the-fly paper prototype.

Advantages: optimization of the information architecture and the navigation

### 5.4.5 Performing the Test

Before beginning the tests, a complete list of questions should be prepared. This makes it possible to estimate which technical coverage is adequate and which form of testing is required. Image for example a WAP application: to evaluate how technical aspects – like different scroll mechanisms, browser versions and display sizes – affect the application studies are necessary, which include different hard- and software. Other questions with little or without reference to technology can be evaluated by using simpler methods like card sorting or paper prototypes. Questions without reference to technology are mostly on the beginning of an application development, like questions of the user’s workflow. Use cases will make it possible to understand how application should work under a given context. Except from the mentioned mobile specific problems the testing itself and also the after-test work is in general not different as for non-mobile devices.
6 Web Usability

6.1 What makes the Web different?

Most of the general principles of user interface design and usability techniques are also applicable to web development, but some things are special when design applications:

- Slower response time, the response times are less predictable than for offline clients.
- The developer has not full control of the appearance of the site; the control is more in the hands of the users and especially the browsers. Some browsers like Mozilla Firefox\(^4\) rely on open and public standards, some browsers more on several proprietary company extensions.
- Main effort when designing web application goes to the design.
- Design tends to kill usability
- Common guidelines and standards very important (e.g. where to put the navigation)

6.2 Guidelines

Web Usability is more than every other software development “obeying the guidelines”. And with guidelines it is like with standards: there are so many…

6.2.1 Design guidelines

In [Niel01] Nielsen describes e.g. 113 Design Guidelines for Homepages; Figure 6-1 shows an adapted checklist with 46 elements:

\(^4\) http://www.mozilla.org/products/firefox/
## Web Technologies - Checklist Homepage Design / Usability

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Topic</th>
<th>Recommended Design</th>
<th>Strength</th>
<th>Points</th>
<th>You</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Download time</td>
<td>(&lt;10 sec for your customer)</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Window title</td>
<td>Start with Company Name</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Title tag line</td>
<td>What about</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Readable URL</td>
<td>Hackable URL</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Error page</td>
<td>Catch errors/dead links, to search</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Meta tags</td>
<td>For search engines</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Alt information</td>
<td>Images, accessibility, Lynx</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Page width</td>
<td>770 pixel (600-1024)</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Liquid vs. frozen layout</td>
<td>Liquid</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Page length</td>
<td>&lt;2 pages (1000-1500 px)</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Frames</td>
<td>No</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Logo placement</td>
<td>Upper left</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Logo size</td>
<td>50x50 Pixel</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Search</td>
<td>Yes, in a box. away</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Search placement</td>
<td>Upper part. right or left corner</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Search box colour</td>
<td>White</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Search button</td>
<td>Call it &quot;Search&quot; or &quot;Go&quot;</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Width of search box</td>
<td>&gt;25 characters (30 best)</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Type of search</td>
<td>Simple search (Link to advanced)</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Navigation</td>
<td>4 types: left, tabs, top, categories</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Footer navigation links</td>
<td>Max. 7 links, single line</td>
<td>*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Sitemap link</td>
<td>Name &quot;Site Map&quot;</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Routing page</td>
<td>No</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Splash page</td>
<td>No</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Sign-In</td>
<td>&quot;Account&quot; or &quot;Sign in&quot;</td>
<td>*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>About the company</td>
<td>Always include it</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>About link</td>
<td>Call it &quot;About &lt;company&gt;&quot;</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Contact information</td>
<td>Call it &quot;Contact us&quot;</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Privacy policy</td>
<td>If you collect data</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Name of privacy link</td>
<td>Call it &quot;Privacy Policy&quot;</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Job opening</td>
<td>Call it &quot;Jobs&quot; if you have it</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Help</td>
<td>If it is a complex site</td>
<td>*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Help placement</td>
<td>Upper right</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Auto-playing music</td>
<td>No</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Animation</td>
<td>No</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Graphical illustration</td>
<td>5-10%</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Advertising</td>
<td>&lt;= 3 ads</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Body text colour</td>
<td>Black</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Body text size</td>
<td>12 points</td>
<td>*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Body text size frozen</td>
<td>No</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Body text typeface</td>
<td>Sans-serif</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Background colour</td>
<td>White</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Link colour (unvisited)</td>
<td>Blue</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Link colour (visited)</td>
<td>Purple</td>
<td>*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Link colour different</td>
<td>Yes (not light grey)</td>
<td>***</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Link underlining</td>
<td>Yes (except in navigation bar)</td>
<td>**</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Score of URL:

100

---

* Default Recommendation
** Strong Recommendation
*** Essential Recommendation

Adapted by Alexander Nischelwitzer, used in [Andr04]

---

Figure 6-1: The Web Usability Checklist, Original from [Niel01] adapted by Alexander Nischelwitzer, used in [Andr04]
6.2.2 Web Accessibility

“Website accessibility is the degree to which a website is accessible to Internet users with disabilities. For example, a web developer might make sure that all images have explanatory captions, so that they can be read by the blind.” [Wiki04e]

It is possible to create websites, which can be visited by e.g. blind people, because the information can be accessed e.g. through a text-only browser\(^5\), screen reader or a Braille device. Common standards for accessibility compliance are the W3C\(^6\) guidelines for creating accessible websites, which are grouped into three levels, Level A—the most basic accessibility—Level AA, and Level AAA.

\(^5\) e.g. Lynx
\(^6\) World Wide Web Consortium, [http://www.w3c.com](http://www.w3c.com)
7 References


[Boer00] Boeree, George; *Gestalt Psychology*; http://www.ship.edu/~cgboeree/gestalt.html; published 2000, visited 06.08.2004

[Davi96] Davis, Stephen Boyd; *The uses of spatiality*; http://www.agocg.ac.uk/reports/virtual/vrmldes/usesa.htm; published June 1996; visited 09.08.2004

[Diam_1] Diamond Bullet; Usability Methods; http://www.usabilityfirst.com/glossary/cat_15.txtl; visited 10.08.2004


[Farm04] Farmer, Arthur; *Usability*; Vienna: CSC Austria internal presentation; 2003

[FMK_1] Forum Mobilkommunikation; *Studie zur volkswirtschaftlichen Bedeutung des Mobilfunsks in Österreich mit ausgewählter Betrachtung der UMTS-Einführung*; www.fmk.at/materialien/download/ACF68D.pdf; visited 10.08.2004
[Gull01]
Gulliksen, Jan; Boivie, Inger; Usability Throughout the Entire Software Development Lifecycle - A Summary of the INTERACT 2001 Workshop; http://www.it.uu.se/research/reports/2001-026/2001-026.pdf; published November 2001; visited 07.09.2004

[HCIS99]
The HCI Space; The Usability Engineering Life Cycle; http://www.tau-web.de/hci/space/x11.html; published 10.03.1999; visited 15.08.2004

[ISO9241]

[Jako_1]
Jacobs, Madeleine; Defining the Difference Between Negative and Positive Painting; http://www.armchairpaintclasses.com/archives/negative-positive.html; visited 06.08.2004

[Litt04]

[Mann01]
Mannhartsberger, Martina; Musil, Sabine; Web Usability - Das Prinzip des Vertrauens; Bonn: Galileo-Press; 2001

[Mayh99]
Mayhew, Deborah J.; The usability lifecycle. A practitioner’s handbook for user interface design; San Francisco: Morgan Kaufmann Publishers, Inc., 1999

[Mayh04]
Mayhew, Deborah J.; DJMA; http://drdeb.vineyard.net/djmasite.pdf; published 2004; visited 14.08.2004

[Niel_1]
Nielsen, Jakob; Ten Usability Heuristics; http://www.useit.com/papers/heuristic/heuristic_list.html; visited 14.08.2004

[Niel_2]
Nielsen, Jakob; Response Times: The Three Important Limits; http://www.useit.com/papers/responsetime.html; visited 12.08.2004

[Niel01]
Nielsen, Jakob; Tahir, Marie; *Homepage Usability: 50 Websites Deconstructed*; Indianapolis: New Riders Publishing; 2001

[Niel03]


[Niel93]

Nielsen, Jakob; *Usability Engineering*; San Francisco: Morgan Kaufmann Publishers, Inc., 1993

[Press92]


[Rask00]

Raskin, Jef; *The humane interface*; Reading, Mass.: Addison-Wesley, 2000

[Thur03]

Thurnher, Bettina; *Usability in Web-Applikationen – Ein Empfehlung für anwenderfreundliche Prozessdarstellungen*; FH Vorarlberg; 2003

[Usab_1]

Usability by Design Ltd.; *Resources – Glossary*; [http://www.usability.uk.com/glossary/glossary.htm](http://www.usability.uk.com/glossary/glossary.htm); visited 10.08.2004

[UsNe03]

UsabilityNet; *International standards for HCI and usability*; [http://www.usabilitynet.org/tools/r_international.htm](http://www.usabilitynet.org/tools/r_international.htm); published 2003; visited 10.08.2004

[Webl03]

Weblab Universität Lüneburg; *Gestaltgesetze*; [http://weblab.uni-lueneburg.de/seminare/webdesign/gestaltgesetze.php](http://weblab.uni-lueneburg.de/seminare/webdesign/gestaltgesetze.php); published 2003, visited 08.08.2004

[Wiki04a]


[Wiki04b]


[Wiki04c]
Wikimedia Foundation Inc.: Usability Testing;  

[Wiki04d]
published 27.08.2004, visited 29.08.2004

[Wiki04e]
Wikimedia Foundation Inc.: Website accessibility;  

[Zuse03]