Usability of the graphical user interface of Android

Bachelor thesis

August 31, 2012

University of Bremen
FB3 Mathematics/Computer Science

Name: Andre Niebuhr
Matriculation number: 2374169
Address: Luisental 29 E
28359 Bremen

Bachelor degree course: Computer Science
Assessor: Prof. Dr. Karl-Heinz Rödiger
Second assessor: Dr. Dieter Müller
I hereby declare that I wrote the following document by myself without using any other resources than mentioned. Literally or correspondingly adopted parts are marked as such and referenced.
Abstract

The spread of smartphones rises rapidly and leads to the faster development of mobile operating systems with more and more functions. Owners of such devices tend to integrate those in their everyday life to attend increasing tasks. But is there enough respect for the needs of users in the design process? This thesis focuses on the usability of the graphical user interface of one of those operating system (Android 2.5.3). Information of the current market situation will be given. Different methods to analyze the usability will be presented and after trading them against each other the chosen method will be used. There will be an explanation for the chosen system as well. In the analysis design weaknesses of the interface will be pointed out and inconsistency in the human computer interaction will be shown. I will convey that there are guidelines and standards which describe good human computer interaction and that even the developers of the system define some. In the conclusion I will show that there are some weaknesses in the system which mainly follow from unthoughtfulness than technical issues. At the end there will be a discussion about the reasons for this circumstance.
Contents

1 Introduction ............................................. 1
  1.1 Motivation ........................................... 1
  1.2 Market situation and state of the art ............... 2
  1.3 Target ............................................... 2
  1.4 General relevance .................................... 3

2 The System ............................................. 4
  2.1 The operating system ................................. 4
  2.2 Limits of consideration ............................... 5

3 Evaluation methods ................................... 6
  3.1 Overview of the methods ............................... 6
    3.1.1 Heuristic evaluation ............................. 7
    3.1.2 Usability testing ................................ 8
    3.1.3 Contextual inquiry ............................... 8
    3.1.4 Logging ......................................... 9
    3.1.5 Cognitive walkthrough ........................... 9
    3.1.6 Expert judgment ................................ 10
  3.2 Chosen method ....................................... 10
  3.3 Evaluation factors ................................... 11

4 Realization ............................................ 13
  4.1 Write a message ..................................... 13
  4.2 Make a call ......................................... 15
  4.3 Search the internet ................................... 16
  4.4 Change the wireless settings ........................ 18
    4.4.1 Wi-Fi ........................................... 19
    4.4.2 Bluetooth ....................................... 21
    4.4.3 Tethering ....................................... 22
    4.4.4 VPN ............................................. 23
    4.4.5 Mobile Networks ................................ 24
  4.5 Add a calendar entry .................................. 26
1 Introduction

This chapter will explain the reason for this thesis. It will introduce you to the main topic and show the questions that are supposed to be answered. The first part will reveal the way I chose the topic. The second part will provide an insight into the current market situation and the state of the art of interaction with mobile media. In the third part the main question of this thesis will be developed and the third part closes the introduction with a description of the general importance of this topic.

1.1 Motivation

I have been using smartphones for about two years now. At the beginning my operating system of choice was Windows mobile 6. There was barely any adaption phase because back then Microsoft made their mobile operating system look similar to Windows XP. After a while the interaction via pen and the resistive touchscreen became annoying and so I decided to change my phone. The Android operation system seemed more suitable for mobile use to me and after a short adaption phase it made using a smartphone a lot easier for me. For the average user this might be the end of the story but since I am studying computer science and I am dealing with user interfaces and mobile devices a lot I noticed that there are a lot of applications for Android systems which are not user friendly and force the user to remember the way to interact after trying everything instead of showing them how. It is easy to become an Android developer and applications do not have to be checked by professionals before they can be released so it is no surprise that there is software developed by amateurs out there. After realizing this I became more attentive to those usability issues when I used the graphical interfaces of the actual operating system. Normally my proposition would be that the operating system has to be really good in matter of usability because it was made by a group of experienced software developers. After paying more attention to usability issues in my everyday use of my smartphone I started wondering if this is really true.
1.2 Market situation and state of the art

In this year the sales figures of smartphones will exceed those of normal mobile phones for the first time due to predictions of the BITKOM organization. The rapid growth of sales figures in the last years lead to faster development of devices and software. While desktop operating systems are usually used for several years, mobile operating systems have got rather short release cycles. Apple (iOS), Microsoft (Windows Mobile) and the Open Handset Alliance (Android) tend to release a new mobile operating system every year. The main reason for those short development periods are the competition between the companies and new technical abilities. Offering new features is a good sales argument. There is one thing all of those mobile operating systems have in common: They all offer graphical users interfaces to enable the user to interact with the device. More than half the amount of the smartphones that are sold this year use the Android operating system. The Open Handset Alliance, founded by Google, released their first operating system with Android, so they have not as much experience in developing operating systems like Apple or Microsoft for example. This fact and the short release cycles of mobile operating systems lead to the question if Android can offer good quality.

1.3 Target

There are very different users of smartphones and not all of them are IT experts which has to be considered when talking about usability of Android systems. This thesis is supposed to provide answer to the questions if Android offers a good usability and is recommendable to all kinds of users. It will be necessary to analyze the graphical user interface of the Android operating system and to rate it with specific factors and with a specific and well suitable method. The positive points as well as the negative ones have to be weighted to come to an answer.

1 http://www.bitkom.org/de/markt_statistik/64086_70921.aspx 17.06.12
2 http://www.gartner.com/it/page.jsp?id=2017015 17.06.12
1.4 General relevance

More and more people are using smartphones and since the Android platform offers a large spread of devices with a large price range it is, at least theoretically, the system with the highest probability to offer a suitable smartphone for everyone. Of major importance on the way to this goal is the usability of those systems. Mistakes in the design of the graphical user interface or missing care of the users needs will, in the long run, lead to a system that people do not want to use. The competitors will take over the dissatisfied customers effortlessly if they keep in mind their needs. Usability engineering is often considered as extensive work with little benefits. The factor ”joy of use” is important when it comes to systems for everyday use and so usability engineering should never be discussed without regard to return on investment. Since smartphone users are not bound to a specific system, like big companies often are when it comes to software, the dissatisfied user will just use another product. Systems with a good usability tend to dissatisfy customers way less than others and that way raise the sales. The significance becomes even more obvious when the improvement of the brands value is considered. Quality is always a good sales argument. The enhancement in value is not just an idealistic point but also visible in numbers. By raising the value and, therefore, the sales and by reducing later changes in the development phase the return for early usability engineering is more than ten times higher than the investment itself. (Marcus 2005) Quality matters a lot when it comes to customer satisfaction. In the requirement analysis for a product there are three types of factors concerning to the Kano model: Basic, performance and excitement. (Nerdinger and Neumann 2007) The basic factors go without saying. They contain features that every customer would expect from a standard product of the kind. If they are fulfilled the customer is neither dissatisfied nor satisfied. They are also called ”must haves”. The performance factors that are especially wanted by the customer and may contain special features. If those are fulfilled the satisfaction of the user rises if not it falls. The excitement factors are those the customer wants but does not explicitly ask for. If they are not fulfilled the satisfaction level is barely lowered but
if they are fulfilled the level rises a lot. It should be noted that a lack of satisfaction does not necessarily mean that the customer is dissatisfied and that a lack of dissatisfaction does not mean the customer is satisfied. The reason for those different influences on the customer satisfaction is that the customer does not always know all features he wants in a product or he can not express his requirements in a way the producer is able to understand properly. Those factors affect all features of a product. If it comes to human computer interaction, however, all features will not satisfy the customer if they are not operable in an appropriate way. That means the interface is the key to create a usable system and therefore the key to quality.

2 The System

This chapter will explain the reasons for the choice of the specific system and emphasize parts which shall not be considered in this thesis.

2.1 The operating system

There are basically two factors that are of importance when choosing the operating system to evaluate: The release date and the amount of devices using it. So the first question is which version of Android is the least recent and is used on most devices. In the case of other systems this would be an easy choice but in the case of Android the two factors diverge. There is a version of Android that is called "Gingerbread". It is installed on over 60 percent of the Android smartphones that are used right now. Then there is a Version called "Ice cream Sandwich" which is installed on less than 17 percent of the smartphones used right now but it is the least recent one.

"Ice cream Sandwich" is the first attempt of the Open Handset Alliance to combine the smartphone operating system and the tablet computer operating system. Though Android is a free software and right now all versions are even open source, the Open Handset Alliance keeps the code of every version

---

3http://de.wikipedia.org/wiki/Android_%28Betriebssystem%29 27.08.12
4http://de.wikipedia.org/wiki/Liste_von_Android-Versionen 27.08.12
Table 1: Detailed description of the used systems

<table>
<thead>
<tr>
<th></th>
<th>Real system</th>
<th>Virtual system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model number</td>
<td>Liquid MT</td>
<td>Acer Laptop</td>
</tr>
<tr>
<td>Android version</td>
<td>2.3.5</td>
<td>2.3.5</td>
</tr>
<tr>
<td>Baseband Version</td>
<td>A4-03.29.01</td>
<td>Unknown</td>
</tr>
<tr>
<td>Kernel version</td>
<td>2.6.35.7-perf</td>
<td>2.6.39.4-android-x86-asus-laptop+cwhuang@tarot)</td>
</tr>
<tr>
<td>Build number</td>
<td>Gingeroumay 2.2.4.000.07</td>
<td>asus_laptop-eng 2.3.5</td>
</tr>
</tbody>
</table>

closed at the point of release. So at first there are some devices which use a new version exclusively. After the code is provided for everyone it takes some time until producers of devices adapt the new version to their devices. This procedure explains why it takes time for new releases to spread. The combination of tablet software and smartphone software makes it even harder to make it work on every device. On the 27th of June a new Android version called ”Jelly Bean” was released. It is at least questionable if ”Ice cream Sandwich” or ”Jelly Bean” will ever reach the spread of ”Gingerbread” and that is why the object of evaluation will be ”Gingerbread” here.

2.2 Limits of consideration

In this thesis only the human computer interaction is regarded. The design and the functionality of graphical user interfaces are the main parts that have to be rated. There is no specific device that is used for the evaluation. At hand there is Acer’s Liquid Metal MT to investigate the functionality on a smartphone in a real life scenario. Also a virtualization via Oracle’s VirtualBox is used to eliminate possible influences of the specific device. The Android Software Development Kit also offers a way to virtualize Android on a computer but first tests showed that it has huge performance problems and could never provide realistic response times. In preparation of this work it is very important to get rid of all the additions to the interfaces that were not implemented by the Open Handset Alliance because producers of
devices tend to change the outer appearance of the operating system before selling it and therefore these additions may have influence on the interaction with the system. Other ways of interaction like speech recognition or gesture recognition stay unattended.

3 Evaluation methods

This chapter will explain the common methods of reviewing human computer interaction. At first it will present an outline of the different methods. Afterwards there will be a detailed description of those methods. The chapter contains the choice of the evaluation method and the reasons why this one is the most suitable and ends with a choice of rating factors.

3.1 Overview of the methods

There are several ways of examining a system. By modifying ways or recombining methods more methods can be created. There are a few approaches that stood the test of time. As explained by Hegner ([Hegner 2003]) the evaluation method to choose depends on the situation. It has to be clear in which state of development the software is, who the tester(s) should be and which information is necessary. Also the correlation of the user, the system and the task have to be kept in mind. A common way to check the usability of software is to let test users use certain parts of the system and interview them after they are done. Instead of the interview a questionnaire is used very often because it is less laborious. In some cases eye tracking is a good choice to see where the users focus is. ([Rudlof 2006]) Participative observation is also a method to collect data about the usability of software. A test users is observed while doing a certain task with the system and after that there can be a short interview with the user to adjust the information taken from the observation with the users impressions. There are two major categories of evaluation methods: descriptive and predictive evaluation. While descriptive methods normally involve test users and clarify the state of the system during the evaluation, the predictive methods need experts and try to point out cer-
tain weaknesses of the usability of the software. (Gediga and Hamborg 2002) Other widely known methods are using log files and verbal protocols. Log files collect data automatically while the software is in use. They can trace the whole application flow the user runs and are able to point out situations in which the user needed above average time to do certain interactions. They also give information about corrections of the user input. The problematic point here is that the collected data have to be interpreted and interpreting the users behavior with nothing more than a list of his actions is quite tough. The data can easily be misinterpreted. The verbal protocol method asks the user to verbalize his thoughts while using the software. Like this certain problems in the interaction sequence can be pointed out and software independent distractions are taken out of the data. However, this method is quite a lot of work. (Henderson et al. 1995) All of these methods have got their advantages and disadvantages and not in every situation every method is suitable. A deeper insight of possibly suitable methods is needed.

3.1.1 Heuristic evaluation

Heuristic evaluation as described by Sarodnick and Brau (Sarodnick and Brau 2006) needs a group of usability experts. They agree on a certain amount of specific heuristics that are used to rate a system. The heuristics describe the desirable behavior of the software and can be derived from guidelines, standards or other documents containing methods of best practice. During the evaluation the experts pretend to be in a real user context. They act like they think normal users would do and pay attention to possible scenarios and user domains. Every aspect that does not match the needs of a single heuristic has to be rated. There is no way of reasoning from a single deflection to a fault in the usability engineering process. In some cases differences to common guidelines can make sense, for example if they avoid larger design mistakes. In other cases the user domain and the scenario rule a perfect match to the heuristics out. The violations of heuristics from all testers have to be compared and categorized to find the source of the problem that leads to mistakes in the interaction. A single usability expert can be very
beneficial in the software development process for the heuristic evaluation, however, one is not enough. For good results there should be at least three experts working together. (Jeffries and Desurvire 1992)

3.1.2 Usability testing

This method needs observers and testers. The test users are asked to execute certain tasks and to say what they think, what they try to do and how they think they can achieve the goal. The observer can either be in the same room as the tester(s) or the testing phase can be recorded and analyzed later. It is indispensable that the observer stays silent during the test phase to avoid disturbing the information flow. After this phase there has to be an interview without a specific structure. The observer tells the participant what he saw and heard just to avoid misunderstandings. The structure of the collected information is created when the interview is over and without the participant. Usability testing offers the possibility to work in a real scenario or in a laboratory environment. In comparison to heuristic evaluation this method can get by with a single expert (the observer) and a random group of participants. (Jeffries and Desurvire 1992) Results may differ for certain groups of participants, so a suitable choice of test users is of vital importance. Another advantage is the missing necessity of filtering the results with regard to real users problems since all problems discovered during this method are problems that would appear in a real scenario. A recorded testing phase also offers the possibility to view certain steps several times and analyze the exact behavior of participants. By splitting test users into groups and giving them different tasks a whole system can be evaluated with few cost.

3.1.3 Contextual inquiry

Contextual interviews require test users with a certain known way of how to do a specific task with the target system. Without a given similar software and users who are familiar with it this method can not be used. The usabil-
ity expert asks the test user to show some of his usual work steps. During this phase the expert assumes the role of a trainee. So questions about the steps are asked. The target is to get to know in which situation the user intends to do what and to see where the user is hindered completing his work. The advantage of this method is that the expert gets to know problems from two sources: The user can talk about problems during the work steps and the expert can see obstacles the user does not notice. ([Beyer and Holtzblatt 1998]) Since this method is based on users experienced with the system it is not quite suitable for analyzing and rating a new system but rather suitable to re-engineer or to replace systems. It can also be used to get a first hint to the question if a change in the system could make sense. Especially when a system is used over a longer period of time and updated regularly this method is useful to check if the development goes the right way.

3.1.4 Logging

To log every interaction with the system is a quite simple way to get data about the usage. While the user interacts with the system all the input data is recorded in log files. ([Hampe-Neteler 1994b]) By adding timestamps it is possible to trace the users way through the system and to see where corrections or delays occurred. One of the down sides of this method is that in a real scenario there is a problem with privacy protection. Also the collected data has to be interpreted. With just the input data given it is most likely that there are misinterpretations. The reason for this is founded in the fact that there can be distractions outside of the system.

3.1.5 Cognitive walkthrough

This method compares different application flows. Initially experts have to think of an optimal way the application should be run through. After this optimal case is set test users will be asked to use the application. Now the user learns how to use it in an explorative way and every difference between the user chosen application flow and the former defined optimal flow
shows steps which have to be reconsidered concerning self-descriptiveness and suitability for learning. (Sarodnick and Brau 2006) Several users will take different ways through the application and if there are overlapping ways from some users these are most likely the most intuitive. Like this developers are able to place parts of the application differently to steer them the way they want to.

3.1.6 Expert judgment

This method is not always standardized. There are two major ways of judgment: The free judgment and the structured judgment. (Hampe-Neteler 1994b) The free judgment offers a simple and fast way of rating a system without former explained criteria. The examination is not further explained and just offers the general impression of the expert. This way the judgment is objective and can not be verified. The structured judgment, however, uses certain rating factors usually taken from known standards. Sometimes also the procedure is explained in the forefront of the examination. Methodically it is often similar to the cognitive walkthrough.

3.2 Chosen method

The expert judgment is a good method in this case because it offers the possibility of including several steps of other methods in a structured way. The chosen factors to evaluate the system are explained in the next chapter. A few tasks will be given to represent usual usage of the system. In addition to these tasks there will be shown special situations which show peculiarities of the system which can not be presented by a single task but rather occur in long term usage. It may not be disregarded that mobile phones are used in a significantly other way than other computers. The graphical user interface is controlled via capacitive touchscreen and so the pointing is not as exact as it would be by using a mouse. Each input element of the user interface has to be controllable by pointing to an area and not to an exact point. Also there is a limited space on the touchscreens which serves as input and output device at once. Another characteristic of mobile phones is that they are barely
switched off. Most users do not turn their devices off ever except for the possible automatic shutdown if the battery runs out of power. Especially the last point will be important later. The tasks that are about to be analyzed are:

1. write a message
2. make a call
3. search the internet
4. change the wireless settings
5. add a calendar entry
6. take a photo

These scenarios were chosen because they involve functions the system offers without using third party software. From my personal experience they also represent the common usage of a smartphone.

3.3 Evaluation factors

In order to achieve a certain amount of measurable results the seven factors of the ISO 9241/110 will be used to rate the user interaction with the system. The seven factors from ISO 9241/110 with a short explanation are shown in table 2.

The single factors can rarely be taken into account without regard to others. In some cases a good solution for one factor can limit the possibility to create a good solution in others. In other cases two or more factors may fortify each other. This is why there might be a need to take several factors into account in one single limited examination while leaving out others. If there are more than one factor to be considered there is a need of checking the weighting of those. Furthermore there is a need of checking the shown problems because some may have bigger influence on the usability than others.
<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suitability for the task</td>
<td>The software has to help the user to accomplish his goal. The Users concentration shall be kept to the task and the system may not disturb this or derail the interaction. No data input that can be generated automatically should be necessary.</td>
</tr>
<tr>
<td>Self-descriptiveness</td>
<td>The user is always aware of what to do next and what an action leads to. System messages have to be easily understandable and explicit.</td>
</tr>
<tr>
<td>Conformity with user expectations</td>
<td>The sequence of interaction steps is like the user would expect it. Consistency in data representation and application flow have to be provided.</td>
</tr>
<tr>
<td>Controllability</td>
<td>The user has to be in control of the application flow. Pausing dialogues has to be possible as well as going back and forth in the application flow at any time.</td>
</tr>
<tr>
<td>Error tolerance</td>
<td>Wrong input data have to be avoided by the input fields. Mistakes in the input data have to be easy to correct. It should be possible to undo actions or if this is not possible (for example deleting data) the user should be warned and should need to confirm it.</td>
</tr>
<tr>
<td>Suitability for individualization</td>
<td>The user should be able to customize certain aspects of the interaction. Appearance, application flow and standard data should be configurable.</td>
</tr>
<tr>
<td>Suitability for learning</td>
<td>The software should enable the user to try new functions. So actions have to be able to be undone. Similar looking things should have the same meaning or function and dialogues have to look the same way.</td>
</tr>
</tbody>
</table>

Table 2: Factors from ISO 9241/110
4 Realization

This chapter represents the actual examination of the interface. The former mentioned scenarios will be executed and concise points will be emphasized. After that the results will be weighted to integrate them into a questionnaire and visualize them like this.

4.1 Write a message

![Figure 1: Writing a message](image)

In order to send a short message or a multimedia message the user can go to the main menu by clicking the menu button on the screen. The process is shown in figure 1. After that the "Messaging" (1.) application can be chosen but this link is also in the home screen. A new screen appears and initially there is nothing to see but a button that says "new message". After a click on it a new screen consisting of three parts appears. The middle part (2.) displays the former messages (and is therefore empty initially), the upper part says "To" (3.) and needs the addressee. Either a phone number can be typed in this field or a name. Typing letters starts a search in the address...
book of the phone and offers a drop down list of matches. The lower field (4.) is the field to type in the actual message. If an input field is chosen the virtual keyboard pops up. In the last step of sending a message the error tolerance is tested by switching of all connections. In this case a notification appears and states, that it is not possible to send the message right now but it will be sent later when there is a connection. By connecting to the network again the message is sent. This task is very intuitively executable without any problems. The limitation of steps in the application do not offer a possibility of differences in the application flow and errors do not disturb the user. The only difficulty is to handle some of the available keyboards because some of them have a ”enter” button (5.) in the lower right corner which leads to a high probability to hit it by accident and send incomplete messages.

Suitability for the task: Makes it possible to send a message of any kind quickly.
Self-descriptiveness: Every input field is described properly and the application flow is easily understandable.
Conformity with user expectations: Application flow is as expected.
Controllability: There are no problems with pausing to write a message or navigating through the steps. The enter button can lead to wrong input data though.
Error tolerance: Errors are shown in an appropriate way and are easily understandable. Correction of the errors are done automatically if possible.
Suitability for individualization: The keyboard can be changed which can be quite useful since there are different types suitable for different persons.
Suitability for learning: Not affected.

Factors lowering the quality of this scenario: Controllability.
4.2 Make a call

Figure 2: The call button on the home screen

![Call Button on Home Screen](image)

Calling a person does not necessarily include the menu since there is a ”Call” button on the home screen as shown in figure 2. There are four sub menus (figure 3) containing favorites, call history, contacts and the phone menu itself. The phone menu shows a virtual numpad to dial the numbers. The contacts menu shows all saved contacts and offers the function to call them or to send a message. After a click on the call button the connection is established. While dialing there are a few options like speaker and mute or ending the call, add call, Bluetooth and showing the numpad. Three of these options are not selectable: Bluetooth, add call and mute. Bluetooth is possibly deactivated while no suitable headset is connected, the reason for deactivating the other two is unknown. Also the function ”add call” is not self-explanatory. If there is no reachable network at the moment an error message is shown. The steps until the actual call are quite intuitive but the
options during the call are not.

Suitability for the task: The interaction is straightforward and there are no unnecessary steps so the suitability for the task is warranted.

Self-descriptiveness: Since there is almost no difference to a conventional phone there is not much to explain.

Conformity with user expectations: The first steps are totally as expected. The deactivated buttons during a call are not explained.

Controllability: Navigation through the few steps is optimal.

Error tolerance: Errors are properly represented and editing wrong input data is easy.

Suitability for individualization: The possibility of dialing via numbers, contacts or favorites lets the user choose his favored way to dial.

Suitability for learning: The limited amount of steps in the application flow does not leave much room to explore the system or alter ways.

Factors lowering the quality of this scenario: Conformity with user expectations.

4.3 Search the internet

![Google search](image)

Figure 4: Searching the internet
Surprisingly searching the internet as shown in figure 4 is the simplest feature of Android phones. The browser button is on the home screen but a Google search can also be executed by showing the keyboard and just typing the words to search. Scrolling in each direction is possible by just pulling the site in the desired direction and the zoom function works by pinching. Further actions are not in the responsibility of Android but in the responsibility of the website developers.

Suitability for the task: Out of the most states of the system it is easily possible to perform a quick internet search.

Self-descriptiveness: Not every way to search is easy to find but since there are easy ones and several ways the self-descriptiveness is quite good.

Conformity with user expectations: Fulfills the expectations.

Controllability: Navigation through the few steps is optimal.

Error tolerance: Wrong input data can easily be corrected and there is always the possibility to go back to the last page or forth to the next like in every desktop internet browser.

Suitability for individualization: Individualization is limited but totally appropriate.

Suitability for learning: Explorative usage gives the user the chance to learn new features of the browser.

Factors lowering the quality of this scenario: None.
4.4 Change the wireless settings

In Order to change the wireless settings the main menu is needed again. Settings is the sub menu of choice here and leads to a menu point called "wireless and networks". A general problem in the settings is that there is no scroll bar visible in the menu unless the user actually scrolls. But without a scroll bar it looks like you already see all the options and you do not automatically try to scroll. The first option here is the "Airplane Mode" which disables all wireless connections. In the virtualized version a click on this field disables the field itself and says "disabling wireless connections" but nothing happens. Even if all connections are turned off via VirtualBox, meaning that there are no possible connections, the airplane mode is available. A click on it does not mark the field or deactivate it. After realizing this I reset the situation on the smartphone. The SIM card was taken out of the phone and other wireless connections were disabled. The airplane mode was still available but the only function it provides now is to make it impossible to turn on Bluetooth while leaving the wireless LAN settings untouched. Under normal circumstances on the smartphone it takes a few seconds and all wireless and mobile connections are disabled.
4.4.1 Wi-Fi

The next option is "Wi-Fi". The wireless LAN can be activated or deactivated here. If the receiver does not work properly it just says "Error". The Wi-Fi settings are right under this field. This new menu begins with a second appearance of the activate Wi-Fi and also behaves exactly the same. It is also possible to set up a notification if an open network is in range. If it is a network that is already known and set up to connect automatically the notification does not appear. Right under this option the already seen wireless networks are listed. On the top of the list there are those in range at the moment. Not currently reachable networks are listed in alphabetic order under the reachables. Only those networks the user attempted to log in to at least once are saved. This list can become quite huge considering how many wireless networks we walk through each day. Deleting those networks from the list has to be done for each network which can take a lot of time. On the bottom there is the option to add a network which is especially important for networks who do not send out an SSID publicly. The menu also offer advanced options like setting a static IP address or power settings for the
wireless LAN. Here it is hard to tell which parts are click-able and which are not because fixed parts like IP address or MAC address are not grey instead they look like the other fields that lead to sub menus. It is also possible to execute a scan for wireless networks but since the system constantly does that it is not needed.

Suitability for the task: Tasks are easily executable.
Self-descriptiveness: Significant and clear nomenclature.
Conformity with user expectations: Options that appear more than once are confusing and white fields that have no input function do not fit into the system.
Controllability: Easily controllable.
Error tolerance: Due to the possibilities no wrong input data is possible. Missing wireless receiver leads to an unclear error message.
Suitability for individualization: The system does not offer the possibility of customization.
Suitability for learning: Limited functionality does not offer the way to alter the handling or discover new functions.

Factors lowering the quality of this scenario: Conformity with user expectations.
4.4.2 Bluetooth

The next part of the wireless settings is Bluetooth. Like before there is a field to activate or deactivate and right under that there is the settings sub menu with the exact same field again. Provoking an error here leads to a different situation than before with the Wi-Fi. In the virtualized Android system there is no Bluetooth receiver available. Clicking on this field leads to a message saying ”Turning on Bluetooth”. It takes a while and the indicator is in the previous state without showing any error message. Other options here are changing the devices name, the visibility, the timeout for the visibility and to scan for other devices. All of those work properly as soon as Bluetooth is turned on. Turning off Bluetooth disables all the other options again and visibility is reset as well as timeout. In contrast to the Wi-Fi settings the Bluetooth settings do not offer advanced options or a manual scan for devices.

Suitability for the task: Tasks are easily executable.
Self-descriptiveness: Significant and clear nomenclature.
Conformity with user expectations: Does not behave like the Wi-Fi settings but looks like it.

Controllability: Easily controllable.

Error tolerance: Wrong input data is avoided by the system but hardware errors are not handled properly.

Suitability for individualization: No possibilities for customization.

Suitability for learning: None.

Factors lowering the quality of this scenario: Conformity with user expectation.

4.4.3 Tethering

The tethering option allows the user to connect a computer to the phone and get an internet connection over the phone or makes the phone act as a wireless hotspot. There is a help section available for the first time. The features work properly. The hotspot settings however offer, like the former wireless settings, again the turning on or off field and an option to configure the hotspot. This configuration is deactivated while the hotspot is turned on so if you enter the menu while it is on you do not get to know why you can
not change settings.

Suitability for the task: Offers just the right amount of information.
Self-descriptiveness: Simple structure and additional help menu explain this part of the system very well.
Conformity with user expectations: The double appearance of the on and off button is again a problem.
Controllability: Offers optimal control.
Error tolerance: No errors could be provoked so the tolerance itself can not be examined.
Suitability for individualization: No individualization possible.
Suitability for learning: The help menu and the link to further information offer the user the possibility to learn more about the functions.

Factors lowering the quality of this scenario: None.

4.4.4 VPN

The virtual private network settings allow you to create a VPN. At first there is the choice of tunneling type. After the type is chosen there is no way to get back. By pressing the back button an error message appears and tells you to choose a name for your VPN first. After that you are forced to
select a server before you are able to get back and so on. The certificates are the biggest problem here because if you just type in wrong data to get back there a no certificates to choose and it is inevitable to quit the menu and start everything from the home screen again.

Suitability for the task: Partially unnecessary steps in order to activate security settings.
Self-descriptiveness: The unitized menu guides the user through the process of adding a VPN step by step.
Conformity with user expectations: Good guidance through the different steps.
Controllability: Impossible to navigate back from a certain point.
Error tolerance: A wrong selected VPN protocol can not be changed and so the user is forced to start over again.
Suitability for individualization: None.
Suitability for learning: No variations of the task and no additional functions to discover.

Factors lowering the quality of this scenario: Suitability for the task, Controllability, Error tolerance.

4.4.5 Mobile Networks

![Figure 10: Mobile Network Settings](image-url)
The mobile networks settings are a sub menu like the above and so the proceeding is like before. Here the mobile internet can be turned on and off as well as the data roaming. A pop up menu allows to set whether to use 2G or 3G networks only or use both, circumstances permitting. Selecting 2G networks only seems to be possible everywhere, selecting 3G only however, is only possible if available and throws an error message like ”network busy” if not. Trying to switch to 3G only several times even makes the process crash.

*Suitability for the task:* Minimal input needed.
*Self-descriptiveness:* Small menu which is easily understandable.
*Conformity with user expectations:* Separation of mobile and wireless networks is not intuitive.
*Controllability:* Changing options and navigating is not a problem.
*Error tolerance:* Network errors are handled in a proper way but there are cases in which the application crashes.
*Suitability for individualization:* None.
*Suitability for learning:* Settings are provided by the service provider and options are not explained so the user is not encouraged to change any of them.

*Factors lowering the quality of this scenario:* Error tolerance, Suitability for learning.
4.5 Add a calender entry

To add an entry to the calender the menu is necessary. The calender button opens a table for the date with 24 slots (hours) to enter data. Clicking one of these slots lets the user create a new appointment. This appointment can be given a name, a date, a duration, a location, a description, participants and reminders in a simple scroll down window. None of these input fields are unconditionally needed but the time. By default the clicked time slot is chosen as a start and the duration is set to one hour. Editing the appointment after saving it is possible without any problems. The typed in participants for the appointment are directly notified via e-mail.

Figure 11: Adding a calender entry

Figure 12: The calender views
Besides the daily view the calendar offers also a weekly view and a monthly one. The scrolling through parts of the calendar is comprehensible after thinking about it. The daily view lets the user scroll through the hours vertically. The horizontal scrolling changes the day. In the weekly view the days are scrolled through horizontally too and the vertical scrolling shows the hours of the seven days. So those two views let the user scroll through days or the larger periods horizontally and through hours or the shorter periods vertically. Of course the monthly view shows a whole month and one would assume that there is either a horizontal scrolling to change the sum of days (which would be a month) like in the weekly view or it would change the larger period which would logically be the year. But neither it is. Actually there is no horizontal scrolling at all in this view and the vertical scrolling changes the month.

Suitability for the task: Just a few steps are necessary to add the entry and all information are optional.

Self-descriptiveness: Every step is self explanatory and the input fields have got default data to tell the user which data should be added.

Conformity with user expectations: Order of actions is very intuitive.

Controllability: Each step can be paused and input data can be edited.

Error tolerance: No errors could be provoked and wrong input data can easily be changed without any loss of information.

Suitability for individualization: Several views and settings for appointments let the user adjust the calendar.

Suitability for learning: Good explanation of all parts of the calendar encourage the user to try new functions.

Factors lowering the quality of this scenario: None.
4.6  Take a photo

![Figure 13: Taking a photo](image)

The camera can be activated by holding the physical camera button. The menu shown in figure 13 on the left appears. The visible scroll bar is used for the zooming function and the buttons (from left to right) make it possible to switch to the video mode, the photo mode, the gallery and it lets the user control the flash settings as well as the automatic focus and illumination setting. The middle part of the figure shows the additional settings which also lead to detailed settings as seen in figure 14. The right part of figure 13 shows the gallery. It offers a comfortable way to scroll through the recorded media and to share or delete one or more files. At the moment there are 368 pictures in the album and it takes a while to load the previews of those. Unfortunately the album does not load previews while idle but loads them just in time while scrolling so the user has to wait every time he scrolls.
“More” offers a bunch of more options concerning the camera. They represent the standard options every digital camera includes. As an unversed photographer like I am the options are not easily understandable but that might be based on the topic. Some options rely on others and get deactivated if others are chosen. For example the anti-shake mode deactivates the ISO settings if turned on because the illumination is able to change the amount of blur in the picture. So these are connected in a logical way. The camera mode and the camera settings do not automatically adapt to the position of the phone. In fact, it is not possible to turn the menu like in the other parts of the system. That is why there are only 4 options visible at once and the user is forced to scroll (the picture is jointed to show all options at once).

*Suitability for the task:* Convoluted menus raise the needed steps to reach certain options.

*Self-descriptiveness:* Not every option is easy to understand though the choice of icons is good.

*Conformity with user expectations:* Turning the phone does not lead to an automatic screen rotation.
Controllability: Visible options can not be selected while others are active. Error tolerance: High performance demands make the application freeze. Suitability for individualization: None. Suitability for learning: The convoluted menus lower the probability that users try new options because they might not even see them.

Factors lowering the quality of this scenario: Suitability for the task, Suitability for learning, Conformity with user expectations, Error tolerance.

4.7 Others

A few abnormal or at least strange states of the system do not appear every time and basically occur in long term usage. This circumstance involves other considerations more than the previous scenarios. The normal usage is not as isolated as the scenarios. The shown parts of the system get mixed in everyday life and other interactions are added or third party applications are used. Also the environment the system is used in differs each time since the device is a mobile one.

There is a list of recently used applications to provide faster access to parts of the system that are used regularly. It offers the possibility to skip unnecessary steps and provides easy access to the favorite applications. But it has a major down side. The application is called in the exact state it was left in the last time. If applications with complex navigation structures are used they show the same state so the usual application flow is not shown which can lead to problems if the user needs an option or window that is not used often. Even advanced users have to search menus to get to some points for this reason. At some points this behavior of the system can get quite confusing. After the scenarios above the recently used applications list is full and there is a button called ”network settings” which leads to the former sub menu of the settings menu. If the back button is hit now it does not open the settings menu as expected but goes back to the home screen so suddenly the network settings are an isolated menu unit. Selecting the settings from
the recently used applications on the other hand leads back to the VPN settings and as already explained there is no way out of this but quitting. The back button has different functions in this case. In the regular settings it brings the user back to the last menu in the network settings it brings the user back to the home screen instead of bringing the user back to the regular settings. The control method is not consistent as well as the dialogue design is not identical which fulfills two criteria of exclusion after Hampe-Neteler at once. (Hampe-Neteler 1994a) A lack of consistency leads to frustration of the user because the system is unable to react like the user would expect it every time (the view in this case is limited to a certain situation) because the same action leads to different reactions. Normally this is easily avoidable and rather occurs because of the inadvertency of the developer and not because of a technical necessity.

During the execution of this evaluation a new aspect attracted my attention which is not directly connected to the graphical user interface but it has influence on its appliance. The problem here is the automatic resource management of the Android operating system. Processor performance and active store are assigned automatically, dynamically and just in time. Most processes of the system can not be quit manually without root rights on the phone and special applications. If the user leaves a menu or an application it is not closed but rather moved to the background. If the actually used application needs more performance or active storage to work properly the operating system just quits another application to provide this performance. The user is not able to control which application gets closed or to set up a priority list. In this case that lead to the following problem: A third party application was needed to take screen shots directly from the phones user interface. This application was set to take a picture each time the accelerometer noticed movement. So the application was started and it was supposed to run in the background waiting for movement. The first few screen shots worked just fine but suddenly changing the menu seemed to need more performance and an application had to be closed. Unfortunately the application to be closed was the screen shot application every time. Like
this I had to restart it in front of every single screen shot. There is also no notification if an application is closed or about which one is closed. In some cases the normal closing process does not work properly and it has to be forced. By forcing the end of the program the state it was in is not saved and an error message pops up.

4.8 Design guidelines

Before finishing the evaluation the question emerges if the Open Handset Alliance does not know the ISO 9241/110. Taking a look into the design guidelines for Android shows that the possibility that they know the standard is quite high. The beginning of the guidelines give a short insight of how an Android application should be designed. After mentioning the creative aspect and the importance of creating an visually attractive application they hand out general information about what to consider. This information represents amongst other things the seven factors from the ISO 9241/110 (the following citations are taken form the website\textsuperscript{5}):

- ”When people use your app for the first time, they should intuitively grasp the most important features.” (Conformity with user expectations)
- ”Android apps empower people to try new things ...” (Suitability for learning)
- ”Android lets people combine applications into new workflows ...” (Suitability for individualization)
- ”It’s not enough to make an app that is easy to use.” (Self descriptive-ness)
- ”Simple tasks never require complex procedures, and complex tasks are tailored to the human hand and mind.” (Suitability for the task)

\textsuperscript{5}http://developer.android.com/design/get-started/creative-vision.html 28.08.12
• ”People of all ages and cultures feel firmly in control, and are never overwhelmed by too many choices or irrelevant flash.” (Controllability, Error tolerance)

Introductory the user gets to know that the size of elements of the user interface is important and there are size ranges for different objects. A button or icon should be between 7mm and 10 mm \(^6\). The space between icons or buttons should be at least 1.5mm . Measuring the sizes on Acer’s Liquid MT I noticed that the keyboard buttons are (at least the alphanumeric ones) 4x6mm and if the phone is held horizontally 5x6mm so the minimal required size is never reached except when it comes to the space between them which is about 2mm. The icons in the main menu are 5x5mm but since the space between them is larger possible accidentally clicks to an icon are counter-vailed.

On another place in the guidelines it is stated that Android menus should not have arrows or similar hints to show that a click lets the user go to a deeper level of the menu \(^7\). As shown in several figures there a a lot of those hints in Android 2.5.3. However, a look into the version 4 of Android shows that those were removed in most parts of the menu.

For some development steps there are even decision diagrams. For example there is a diagram to help the developer to decide whether something should be a setting or not \(^8\). It is implied that often changed values should rather not be made a setting but be handled differently. That means they have to be accessible with less than three actions, concerning to the guideline. As shown before the wireless LAN or Bluetooth settings need four steps to be accessed. In Android 4 the developers improved this but it still takes three actions to access the settings.

\(^6\)http://developer.android.com/design/style/metrics-grids.html 28.08.12
\(^7\)http://developer.android.com/design/patterns/pure-android.html
\(^8\)http://developer.android.com/design/patterns/settings.html
Even though this was just a very short insight into the design guidelines of the Android Open Source Project it becomes clear that there are certain guidelines which are sometimes a little vague but sometimes very specific. Although they are freely accessible for everyone not even the Open Handset Alliance fully sticks to them.

5 Conclusion

The previous chapter mainly points out negative parts of the system. As already explained the reason for this is that mistakes in the basic concept of the usage attract way more attention than good solutions. Good design does not affect the quality as much in the positive direction as bad one does in the negative direction. Nevertheless, this does not automatically mean that Android 2.5.3 is a bad system. To visualize the results of this work there is a questionnaire in the appendix. It represents the information pointed out in the previous chapter with respect to the good designed parts that did not need to be emphasized in a special way. Also the implied weighting in the chapters of the separate factors are taken into account.

The questionnaire was taken from the course "Interaktive Systeme 1" (winter term 2010/2011) and, as stated on it, made by Prümper and Anft. The field concerning the weekly hours of work was left empty because it is insignificant in this case. As shown in the questionnaire the suitability for the tasks was pretty good. The only negative point in these questions was about automating reoccurring tasks which is barely possible. Most of the time, especially when it comes to standard tasks, the system keeps the application flow short. The self descriptiveness of the system is rather mediocre to bad. The possibilities are not directly noticeable and so users might oversee benefits of the functions. Also the given information and help is often deficient. Since I do not think that self descriptiveness is the most important factor in a system with short application flows like this, these points do not influence my evaluation as much as others. The conformity with the user expectations is quite intermingled. Concurrent tasks have a major influence on the system
behavior so that most of the time the expectations fit to the application flow but in other cases when there are more processes, there are freezes in the application or they crash. The suitability for learning is mediocre to good. The positive influence results from the short ways between the applications which offer the user the possibility to work by trial and error. The negative parts mainly result from the partially strong convoluted menu structures which make it hard to find new options. Controllability of the system is very good. The user is normally in total control of the application flow and can switch windows, pause steps or go back through menus. The error tolerance is the most difficult part to evaluate. The user is always informed immediately if he entered something wrong or data is missing. On the other hand errors lead to application crashes which should not happen at all. Since separated actions in the system are quite limited in their functionality there are almost no system immanent possibilities to customize the system. Nevertheless, Android systems are highly modular constructed and so most parts of the system can be exchanged for alternatives and third party implementations of the same or similar functions.

The operating system is obviously not perfect. But it is not that bad either. There are parts of the system which are designed very good. They offer information of how to use the system and explain which input will lead to which reaction. The application flow can be interrupted without loss of data and can be continued later. No unnecessary data is needed and errors are avoided by the way the data is collected. There is almost exemplary design to be found in those parts. Unfortunately, this design is not consistently to be found in the whole system.

Since all versions of the Android operating systems are highly customizable for almost every problem there is a solution out there. If this thesis would deal with a customized version of Android it might be much more positive since there are a lot of versions created by competent users who see some of the issues with the official system. Since the issues are not fixed in the official versions one can assume that the Android developers do not seize the
chance to look into user created adaptions and therefore they do not use the cheapest way of usability evaluation. Since custom operating systems are very popular the quality or functionality of those have to be an improvement to the official versions. The negative sides of user customized operating systems is the fact that they are not always up to date and that there is no guarantee that those are stable versions.

So there are a lot of points Android needs improvement in. Some of them are of minor importance some others are quite important. A short outlook on the version 4 of Android which I made during my research showed that there is a certain development but some of the mistakes are just kept up and other new ones are added while some are removed. The worst point of all this is that there would be a way to make Android a great mobile operating system. It would just take a little more effort on usability. Analyzing popular custom versions of released systems are like a free usability test from a large crowd of people. The mistakes concerning the consistency could even be avoided easier by an improvement in the usability management in the forefront. The Open Handset Alliance seems to set their priorities more to releasing new systems than to releasing and evolving one really good systems.

6 Summary and future prospects

If the the software development has reached the phase where it is not the obstacle to solve a problem technically anymore there will be different solutions and factors like usability and joy of use become important. For this reason this thesis dealt with the usability of the Android graphical user interface. It was explained that usability is a major factor to provide quality and that quality in the long run raises value and therefore leads to a good return on investment. As shown in the research there are a lot of ways the usability of software can be evaluated and most known ways are not even that new so that it is assumable that some people recognized the meaning of user interfaces for a software’s value a while ago. It was also obvious that different methods are only suitable for certain situations and the choice of the method
and the test user(s) can have a big influence on the results. The evaluation showed that the consistency of the operating system is sometimes not given. While parts of the system are not operable in a good way other parts, in fact quite similar parts, are handled far better. It is not comprehensible why the different parts of the system are not handled in the same way. The short insight to the Android design guidelines also show that there are rules which shall be followed when it comes to Android development but this just emphasizes the impression that the developers of the operating system do not attend those enough. Since the different parts of the system differ so much in quality it is logical to presume that the modules of the software are worked on by different teams in the development process. Some of these teams seem to keep more respect to usability and the Android Open Source Project’s guidelines than others. The fact that there are faults in the interface design which are not corrected by updates or in later versions show that usability is, at least partially, neglected. Putting more effort in the usability engineering at an early point of the development could increase the quality of the system a lot. It could also prevent attacks on the system’s reputation like the claim Android could only be used by computer scientists, which Microsoft’s Steve Ballmer once expressed[^1]. A consistent design with a high usability could also have major influence on third party software. If a developer uses a system and intents to expand it or to develop software for it, the developer will analyze the system first. If the interfaces are consistent the developer will, maybe not even knowingly, most likely tend to follow the guidelines. If the developer on the other hand does not notice a clear and consistent structure in the graphical user interface he will create one himself. This leads to a lot of different operation structures. In other words the user of those products has to learn how to handle an application each time he uses a new one. Following guidelines would make it easier to use new applications and lower the adaption phase. An evolution in this direction could lead to a whole new level of quality in Android applications and improve the reputation of it. New customer groups could be reached which would lead to a lot of pressure on the competitors. Competitors under pressure are forced to improve their

[^1]: http://winfuture.de/news,66110.html 29.08.12
products by offering more quality or individual innovations. Advancement in usability might drive forward the technical progress of mobile media and if the Open Handset Alliance does not take that chance, someone else will.
References


**Internet references**

Android Open Source Project


BITKOM

- [http://www.bitkom.org/de/markt_statistik/64086_70921.aspx](http://www.bitkom.org/de/markt_statistik/64086_70921.aspx) 17.06.12

Gartner

- [http://www.gartner.com/it/page.jsp?id=2017015](http://www.gartner.com/it/page.jsp?id=2017015) 17.06.12

Wikipedia

- [http://de.wikipedia.org/wiki/Android_%28Betriebssystem%29](http://de.wikipedia.org/wiki/Android_%28Betriebssystem%29) 27.08.12

Winfuture

- [http://winfuture.de/news,66110.html](http://winfuture.de/news,66110.html) 29.08.12
List of Figures

1. Writing a message ........................................ 13
2. The call button on the home screen ....................... 15
3. The call menu .............................................. 15
4. Searching the internet ..................................... 16
5. Wireless and Networks .................................... 18
6. WiFi Settings ............................................. 19
7. Bluetooth Settings ....................................... 21
8. USB/Tethering Settings ................................... 22
9. VPN settings ............................................. 23
10. Mobile Network Settings ................................ 24
11. Adding a calendar entry ................................ 26
12. The calendar views ...................................... 26
13. Taking a photo ........................................... 28
14. Camera settings .......................................... 29

List of Tables

1. Detailed description of the used systems .................. 5
2. Factors from ISO 9241/110 ................................ 12
A ISO questionnaire

ISONORM 9241/110 (Langfassung)
Beurteilung von Software auf Grundlage der Internationalen Ergonomie-Norm DIN EN ISO 9241-110

von Prof. Dr. Jochen Prümper und Dipl. Psych. Michael Anft

Im Folgenden geht es um die Beurteilung von Softwaresystemen auf Grundlage der Internationalen Norm ISO 9241/110.

Bitte beachten Sie:

- Das Ziel dieser Beurteilung ist es, Schwachstellen bei Softwaresystemen aufzudecken und konkrete Verbesserungsvorschläge zu entwickeln.
- Um dies zu bewerkstelligen, ist ihr Urteil als Konner des Softwaresystems von entscheidender Bedeutung! Grundlage Ihrer Bewertung sind Ihre individuellen Erfahrungen mit dem Software-Programm, das Sie beurteilen möchten.
- Dabei geht es nicht um eine Beurteilung Ihrer Person, sondern um Ihre persönliche Bewertung der Software mit der Sie arbeiten.

Bitte machen Sie im folgenden Kasten zunächst einige Angaben zu der Software, auf die sich Ihre Beurteilung im Folgenden beziehen wird.

<table>
<thead>
<tr>
<th>Auf welches Software-Programm bezieht sich Ihre Beurteilung?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Beurteilen Sie bitte lediglich ein Software-Programm!)</td>
</tr>
<tr>
<td>Name der Software</td>
</tr>
<tr>
<td>Android</td>
</tr>
<tr>
<td>Versionsnummer</td>
</tr>
<tr>
<td>2.3.5</td>
</tr>
<tr>
<td>Hersteller</td>
</tr>
<tr>
<td>Open Handset Alliance</td>
</tr>
<tr>
<td>gegebenenfalls Teilanwendung / Modul</td>
</tr>
<tr>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>Wie hoch ist Ihre wöchentliche Arbeitszeit?</td>
</tr>
<tr>
<td>Stunden pro Woche</td>
</tr>
<tr>
<td>Wie viele Stunden arbeiten Sie pro Woche durchschnittlich mit Bildschirmgeräten?</td>
</tr>
<tr>
<td>70 Stunden pro Woche</td>
</tr>
<tr>
<td>Wie viele Stunden arbeiten Sie pro Woche durchschnittlich mit der von Ihnen beurteilten Software?</td>
</tr>
<tr>
<td>10 Stunden pro Woche</td>
</tr>
</tbody>
</table>
Seit wie vielen Monaten arbeiten Sie schon mit der von Ihnen beurteilten Software?

| Monate | 14 |

Wie gut beherrschen Sie die von Ihnen beurteilte Software?

| - | - | - | - | - | - | - | +++ |

(--- = sehr schlecht; +++ = sehr gut)

--- --- -/- + + +++

--- --- -/- + + +++

Noch ein Hinweis zur Beantwortung des Beurteilungsbogens:

Im folgenden Fragebogen werden die Anforderungen der Norm über Beschreibungen konkretisiert. Diese Beschreibungen weisen immer folgende Form auf:

Beispiel 1:

| Die Software ... | --- | - | -/+ | + | ++ | +++ | Die Software ... |
|-----------------|-----|---|------|---|----|-----|
| ist schlecht.    | ☐   | ☐ | ☐    | ☒ | ☐  | ☐   |
| ist gut.         | ☒   | ☐ | ☐    | ☐ | ☐  | ☐   |

Im ersten Beispiel wird danach gefragt, wie gut bzw. wie schlecht die Software ist. Die Benutzerin oder der Benutzer beurteilt in diesem Fall die Software zwar als gut, sieht jedoch noch Verbesserungsmöglichkeiten.

Beispiel 2:

| Die Software ... | --- | - | -/+ | + | ++ | +++ | Die Software ... |
|-----------------|-----|---|------|---|----|-----|
| ist langsam.    | ☒   | ☐ | ☐    | ☒ | ☐  | ☐   |
| ist schnell.     | ☐   | ☐ | ☐    | ☒ | ☐  | ☐   |

Im zweiten Beispiel beurteilt die Benutzerin oder der Benutzer die Software als ziemlich langsam.

Am Ende jeder Seite werden Sie gefragt, wie wichtig Sie das jeweilige Norm-Kriterium für Ihre Tätigkeit finden.

In einem weiteren Feld auf jeder Seite können Sie ein Beispiel angeben, welches die Verletzung des jeweiligen Norm-Kriteriums konkretisiert.

- Am besten bearbeiten Sie den Beurteilungsbogen, während Sie das zu bewertende Softwaresystem vor sich am Bildschirm haben. Dadurch haben Sie die Möglichkeit, bei der Beantwortung der einzelnen Fragen die eine oder andere Sache noch einmal zu überprüfen.

- Füllen Sie bitte den Beurteilungsbogen äußerst sorgfältig aus und lassen Sie keine der Fragen aus!

Beurteilung der Software – 2/9
### Aufgabenangemessenheit

Unterstützt die Software die Erledigung ihrer Arbeitsaufgaben, ohne Sie unnötig zu belasten?

<table>
<thead>
<tr>
<th>Die Software ...</th>
<th>- - - -</th>
<th>-/+</th>
<th>++</th>
<th>+++</th>
<th>Die Software ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ist kompliziert zu bedienen.</td>
<td>○ ○ ○ ○</td>
<td>X</td>
<td>○ ○ ○ ○</td>
<td>ist unkompliziert zu bedienen.</td>
<td></td>
</tr>
<tr>
<td>bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.</td>
<td></td>
</tr>
<tr>
<td>bietet schlechte Möglichkeiten, sich häufig wiederholende Bearbeitungsvorgänge zu automatisieren.</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>bietet gute Möglichkeiten, sich häufig wiederholende Bearbeitungsvorgänge zu automatisieren.</td>
<td></td>
</tr>
<tr>
<td>erfordert überflüssige Eingaben.</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>erfordert keine überflüssigen Eingaben.</td>
<td></td>
</tr>
<tr>
<td>ist schlecht auf die Anforderungen der Arbeit zugeschnitten.</td>
<td>○ ○ ○ ○</td>
<td>○ ○ ○ ○</td>
<td>X</td>
<td>ist gut auf die Anforderungen der Arbeit zugeschnitten.</td>
<td></td>
</tr>
</tbody>
</table>

### Aufgabenangemessenheit der Software ist für meine Tätigkeit ...

| sehr unwichtig | ○ ○ ○ ○ ○ | X | ○ ○ ○ ○ | sehr wichtig |

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht.

Chapter with significant influence on this part: 4.6

---

Beurteilung der Software – 3/9
### Selbstbeschreibungsfähigkeit

Gibt Ihnen die Software genügend Erläuterungen und ist sie in ausreichendem Maße verständlich?

<table>
<thead>
<tr>
<th>Die Software ...</th>
<th>+++</th>
<th>++</th>
<th>+</th>
<th>-</th>
<th>-</th>
<th>Die Software ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>bietet einen schlechten Überblick über Ihr Funktionsangebot.</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>bietet einen guten Überblick über Ihr Funktionsangebot.</td>
</tr>
<tr>
<td>verwendet schlecht verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>verwendet gut verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.</td>
</tr>
<tr>
<td>liefert in unzureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>liefert in ausreichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.</td>
</tr>
<tr>
<td>bietet auf Verlangen keine situationsspezifischen Erklärungen, die konkret weiterhelfen.</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>bietet auf Verlangen situationsspezifische Erklärungen, die konkret weiterhelfen.</td>
</tr>
<tr>
<td>bietet von sich aus keine situationsspezifischen Erklärungen, die konkret weiterhelfen.</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>⬤</td>
<td>bietet von sich aus situationsspezifische Erklärungen, die konkret weiterhelfen.</td>
</tr>
</tbody>
</table>

Die Selbstbeschreibungsfähigkeit der Software ist für meine Tätigkeit ...

| sehr unwichtig | ⬤ | ⬤ | ⬤ | ⬤ | ⬤ | ⬤ | sehr wichtig |

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht:

---

Chapters with significant influence on this part: 4.6, 4.3

---

Beurteilung der Software – 4 / 9
**Erwartungskonformität**

Kommt die Software durch eine einheitliche und verständliche Gestaltung Ihren Erwartungen und Gewohnheiten entgegen?

<table>
<thead>
<tr>
<th>Die Software ...</th>
<th>- - - - + + + + -</th>
<th>Die Software ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>erschwert die Orientierung durch eine uneinheitliche Gestaltung.</td>
<td>erschwert die Orientierung durch eine uneinheitliche Gestaltung.</td>
<td></td>
</tr>
<tr>
<td>lässt einen im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.</td>
<td>lässt einen im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.</td>
<td></td>
</tr>
<tr>
<td>informiert in unzureichendem Maße über das, was es gerade macht.</td>
<td>informiert in ausreichendem Maße über das, was es gerade macht.</td>
<td></td>
</tr>
<tr>
<td>reagiert mit unvorhersehbaren Bearbeitungszeiten.</td>
<td>reagiert mit gut vorhersehbaren Bearbeitungszeiten.</td>
<td></td>
</tr>
<tr>
<td>lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen.</td>
<td>lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.</td>
<td></td>
</tr>
</tbody>
</table>

Die Erwartungskonformität der Software ist für meine Tätigkeit ...

<table>
<thead>
<tr>
<th>sehr unwichtig</th>
<th>sehr wichtig</th>
</tr>
</thead>
</table>

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht:

**Chapters with significant influence on this part: 4.2, 4.4, 4.6**

Beurteilung der Software – 5/9
### Lernförderlichkeit

Ist die Software so gestaltet, dass Sie sich gut darin einarbeiten konnten und bietet sie auch dann Unterstützung, wenn Sie neue Funktionen lernen möchten?

<table>
<thead>
<tr>
<th>Die Software...</th>
<th>+++</th>
<th>++</th>
<th>+</th>
<th>-/+</th>
<th>++</th>
<th>+++</th>
<th>Die Software...</th>
</tr>
</thead>
<tbody>
<tr>
<td>schw16</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td>☒</td>
<td>erfordert viel Zeit zum Erlernen.</td>
</tr>
<tr>
<td>schw17</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>ermutigt dazu, auch neue Funktionen auszuprobieren.</td>
</tr>
<tr>
<td>schw10</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>erfordert, dass man sich viele Details merken muss.</td>
</tr>
<tr>
<td>schw15</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>ist so gestaltet, dass sich einmal Gelerntes schlecht einprägt.</td>
</tr>
<tr>
<td>schw20</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>ist schnell ohne fremde Hilfe oder Handbuch erlernbar.</td>
</tr>
</tbody>
</table>

Die Lernfähigkeit der Software ist für meine Tätigkeit...

| sehr unwichtig | ☐ | ☐ | ☐ | ☒ | ☐ | ☒ | sehr wichtig |

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht:

*Chapters with significant influence on this part: 4.4, 4.6*

Beurteilung der Software – 6/9
### Steuerbarkeit

Können Sie die Art und Weise, wie Sie mit der Software arbeiten, beeinflussen?

<table>
<thead>
<tr>
<th>Die Software bietet keine Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und dort später ohne Verluste wieder weiterzumachen.</th>
<th>Die Software bietet die Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und dort später ohne Verluste wieder weiterzumachen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw21</td>
<td>sw21</td>
</tr>
<tr>
<td>erzwinge eine unnötig starre Einhaltung von Bearbeitungsschritten.</td>
<td>erzwinge keine unnötig starre Einhaltung von Bearbeitungsschritten.</td>
</tr>
<tr>
<td>sw22</td>
<td>sw22</td>
</tr>
<tr>
<td>ermöglicht keinen leichten Wechsel zwischen einzelnen Menüs oder Masken.</td>
<td>ermöglicht einen leichten Wechsel zwischen einzelnen Menüs oder Masken.</td>
</tr>
<tr>
<td>sw23</td>
<td>sw23</td>
</tr>
<tr>
<td>ist so gestaltet, dass der/die Benutzer/in nicht beeinflussen kann, wie und welche Informationen am Bildschirm dargeboten werden.</td>
<td>ist so gestaltet, dass der/die Benutzer/in beeinflussen kann, wie und welche Informationen am Bildschirm dargeboten werden.</td>
</tr>
<tr>
<td>sw24</td>
<td>sw24</td>
</tr>
<tr>
<td>erzwinge unnötige Unterbrechungen der Arbeit.</td>
<td>erzwinge keine unnötigen Unterbrechungen der Arbeit.</td>
</tr>
</tbody>
</table>

**Die Steuerbarkeit der Software ist für meine Tätigkeit...**

<table>
<thead>
<tr>
<th>sehr unwichtig</th>
<th>sehr wichtig</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw25</td>
<td>sw25</td>
</tr>
</tbody>
</table>

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht:

---

*Chapter with significant influence on this part: 4.1*

---

Beurteilung der Software – 7 / 9
### Fehlertoleranz

Bietet Ihnen die Software die Möglichkeit, trotz fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis ohne oder mit geringem Korrekturaufwand zu erreichen?

<table>
<thead>
<tr>
<th>Die Software ...</th>
<th>+++</th>
<th>++</th>
<th>+</th>
<th>0</th>
<th>-</th>
<th>+++</th>
<th>Die Software ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ist so gestaltet, dass kleine Fehler schwerwiegende Folgen heben können.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ist so gestaltet, dass kleine Fehler keine schwerwiegenden Folgen haben können.</td>
</tr>
<tr>
<td>informiert zu spät über fehlerhafte Eingaben.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>informiert sofort über fehlerhafte Eingaben.</td>
</tr>
<tr>
<td>liefert schlechte verständliche Fehlermeldungen.</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>liefert gut verständliche Fehlermeldungen.</td>
</tr>
<tr>
<td>erfordert bei Fehlern im Größen und Ganzen einen hohen Korrekturaufwand.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>erfordert bei Fehlern im Größen und Ganzen einen geringen Korrekturaufwand.</td>
</tr>
<tr>
<td>gibt keine konkreten Hinweise zur Fehlerrbehebung.</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gibt konkrete Hinweise zur Fehlerrbehebung.</td>
</tr>
</tbody>
</table>

Die **Fehlertoleranz** der Software ist für meine Tätigkeit...

- sehr unwichtig  
  ![Image](image-url)
  - sehr wichtig

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht:

**Chapters with significant influence on this part: 4.4, 4.6**

Beurteilung der Software – 8/9
### Individualisierbarkeit

Können Sie als Benutzer oder Benutzerin die Software ohne großen Aufwand auf ihre individuellen Bedürfnisse und Anforderungen anpassen?

<table>
<thead>
<tr>
<th>Die Software lässt sich von mir schwer erweitern, wenn für mich neue Aufgaben entstehen.</th>
<th>+++</th>
<th>++</th>
<th>+/-</th>
<th>++</th>
<th>+++</th>
<th>Die Software lässt sich von mir leicht erweitern, wenn für mich neue Aufgaben entstehen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>lässt sich von mir schlecht an meine persönliche, individuelle Art der Arbeitserledigung anpassen.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lässt sich von mir gut an meine persönliche, individuelle Art der Arbeitserledigung anpassen.</td>
</tr>
<tr>
<td>eignet sich für Anfänger und Experten nicht gleichermaßen, weil ich es nur schwer an meinen Kenntnisstand anpassen kann.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>eignet sich für Anfänger und Experten gleichermaßen, weil ich es leicht an meinen Kenntnisstand anpassen kann.</td>
</tr>
<tr>
<td>ist so gestaltet, dass ich die Bildschirmdarstellung schlecht an meine individuellen Bedürfnisse anpassen kann.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ist so gestaltet, dass ich die Bildschirmdarstellung gut an meine individuellen Bedürfnisse anpassen kann.</td>
</tr>
</tbody>
</table>

### Die Individualisierbarkeit der Software ist für meine Tätigkeit ...

| sehr unwichtig |       |       |       |       |       |       | sehr wichtig |

Bitte beschreiben Sie ein Beispiel, welches die Verletzung der oben genannten Aspekte gegebenenfalls besonders deutlich veranschaulicht:

This part is a special one in this context because the system itself does barely provide any possibility for individualization but there a a lot of third party applications and modified versions which offer a high individualization. Both points were taken into account.

---

Beurteilung der Software – 9/9