University of Würzburg
Institute of Computer Science
Research Report Series

A Survey on Usability Evaluation Techniques and an Analysis of their actual Application

Martina Freiberg, Joachim Baumeister

Report No. 450 October 2008

University of Würzburg
Institute of Computer Science
Department of Artificial Intelligence and Applied Informatics
Am Hubland, D-97074 Würzburg, Germany
{freiberg,baumeister}@informatik.uni-wuerzburg.de

A Survey on Usability Evaluation Techniques and an Analysis of their actual Application

Martina Freiberg, Joachim Baumeister
University of Würzburg
Institute of Computer Science
Department of Artificial Intelligence and Applied Informatics
Am Hubland, D-97074 Würzburg, Germany
{freiberg,baumeister}@informatik.uni-wuerzburg.de

Abstract

Today, software products for nearly every possible purpose exist. Most of them enable users to accomplish their tasks somehow, yet often they do not support them in doing so. In many cases this is either due to the complex design of the software interfaces, or to a poorly designed underlying task model, leading to time-consuming procedures.
This is why interface design, especially in terms of ergonomics and usability, is still a fruitful field of research, providing many topics of interest for applied scientific works as, for example, Ph.D. dissertations. Likewise growing is the field of interface and usability evaluation, already providing a large number of different techniques. This raises the need for a comprehensive categorization of the techniques. Another interesting research question is, which techniques are actually usable and have been applied in the context of scientific theses so far.

This research report contributes to these topics in providing an extensive, categorized overview of usability evaluation techniques, and furthermore by reporting the results of the analysis of their actual application in the named context. Therefore, the relevant literature was reviewed and various Ph.D. and MA theses from computer science and strongly related fields from 2000 to 2008 (incl.) were collected and analyzed.

Key words: Human Computer Interation, Usability Evaluation, User Interface Evaluation

1 Introduction

With the increasing capacity of today's computers, the number of software products for nearly every possible purpose also grows steadily. Most of them enable users to accomplish their tasks. However, too many are difficult to use and require expert knowledge or extensive training. Sometimes, even proficient users repeatedly struggle with their usage. In many cases, this is due to the complexity and poor design of the software interfaces. These often do not provide adequate support for the users, but worse, some even hinder them to work efficiently. "Joy of use" is only achievable by very few pieces of software. Moreover, today's software often contains at least some time-consuming tasks, often due to an underlying poor task design, resulting in complex procedures and/or long system response times. As it is crucial for most software users to achieve their tasks as quickly and straightforward as possible, many give up after some time, if not entirely refuse, using software consisting of complicated tasks and long-lasting procedures.
This is the reason, why research on interface design, especially in terms of ergonomics and usability, is still a fruitful field. Various recommendations, guidelines and standards on designing usability-conforming software interfaces have been proposed and several are already widely accepted among developers and interface designers. Consequently, also a lot of different techniques for evaluating interfaces in terms of usability have been suggested to date, including not only the design of a user interface (that is, aesthetical aspects), but also the underlying task design (that is, functionality-related aspects). The amount of existing distinct usability evaluation methods raises the need for a comprehensive categorization of the techniques. Due to their topicality, interface design and

usability evaluation provide various interesting topics to be addressed by scientific theses. This subsequently poses the question, which techniques are actually usable and have been applied in the context of scientific theses so far.

To contribute to this issues, we first provide a comprehensive, categorizing overview of existing usability evaluation techniques. Surveys of this kind have already been presented by several researchers. Hegner [58] and Stowasser [144], for example, present quite extensive overviews of usability evaluation techniques. More focussed surveys are provided by Hollingsed & Novic [61] (expert evaluation), Nielsen [114] (expert evaluation), and three chapters of the Human-Computer Interaction Handbook [69] by Jacko & Sears: chapter 56 (user-based evaluation [39]), 57 (expert evaluation [23]) and 58 (model-based evaluation [81]). As more, many books, covering the broader topics of interface design or the usability engineering process, also provide surveys on evaluation techniques. Examples that served as a helpful basis for this report are: "Usability Engineering" (Nielsen & Mack [114]), "Designing the User Interface" (Shneiderman & Plaisant [139]), "The Essential Guide to User Interface Design" (Gallitz [46]), "Human-Computer Interaction" (Dix et al. [35]), "Usability Engineering" (Leventhal & Barnes [86]), "Human Computer Interaction" (Te'eni et al. [148]), "User Interface Design and Evaluation" (Stone et al. [143]) and "Interaction Design" (Sharp et al. [138]). The main difference between those surveys and book chapters on usability evaluation and the present work is, that we additionally investigated the applicability and actual applicance of the presented techniques in recent scientific works. Thus we hope to reveal the relation between the theoretic approaches and their practicability in reality. Another difference is, that—due to the lively research interest and ongoing developments in this field—some of those previous works do not cover all of today's known techniques. Moreover, some researchers intentionally focus their survey on just one of the two main categories (user-based and expert evaluation). In contrast to that, we tried to cover all main usability evaluation techniques known today.

To address the question, which of the categorized techniques are actually practicable and have been applied in scientific theses today, appropriate works were collected and analyzed. Thereby we focused specifically on Ph.D. and MA theses, mainly from computer science or strongly related fields. To provide topicality, only more recent works from the years 2000 to 2008 were examined. Based on the analysis, the actual application of evaluation techniques is described. To collect appropriate works, we consulted public libraries as well as both nationwide and international online services as for example the OAIster, OASE or the DIVA portal. Applied search criteria included, for example, "interface evaluation", "usability evaluation", "interface ergonomics and evaluation", and similar terms related to *HCI* (Human Computer Interaction) and usability evaluation.

The paper is organized as follows: Section 2 presents an overview of usability evaluation techniques known to date. These methods are classified into three categories, as shown in Figure 1. Each technique is summarized and literature for further information is provided. Section 3 presents the results of the analysis of the collected theses. Therefore, a synoptical table (Table 5) is provided in section 3.1, summing up general information on each work and on the applied evaluation techniques. Also table parameters, that is, the columns and the possible entry types, are listed and shortly explained in the same section, to enable the reader to better understand the table data. Section 3.2 provides a detailed analysis of the most interesting findings of our research study. Moreover, a complete listing of the sources—that is, basic literature covering fundamental theories and procedures the researchers of the examined works applied to implement their own specific forms of evaluation—is provided. Finally, section 4 summarizes the main findings and provides a short conclusion.

2 Usability Evaluation Techniques

To survey currently known usabilty evaluation techniques, we classified them according to three main categories in terms of the type of evaluator: user-based evaluation (section 2.1), expert evaluation (section 2.2), and hybrid approaches (section 2.3), as depicted in Figure 1. The latter consists of five techniques, that cannot be assigned clearly to the former two categories, because

of two reasons: first, the *collaborative usability inspection*, the *participatory heuristic evaluation*, and the *pluralistic walkthrough technique* are the result of merging user-based and expert evaluation approaches; second, the other two techniques—*competitve/comparative analysis* and *usability evaluation of web-sites*—can be conducted on the basis of user-based and expert evaluation techniques.

As the purpose of this section is to present a comprehensive overview, the techniques are in the following not only summarized each, but also references for more detailed and further reading are provided. Subsequent to the description of the categorized techniques, section 2.4 presents some additional methods—not specifically intended for usability evaluation—that nevertheless can enhance usability evaluation, if applied supplementary.

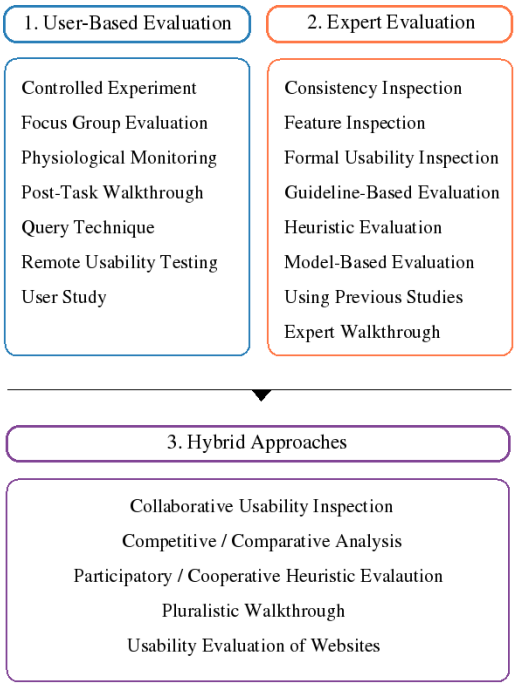

Figure 1: Overview of Evaluation Methods

2.1 User-based Evaluation

Common ground of all user-based evaluation techniques is the necessary participation of representative target users. The main advantage is the possibility to directly explore the user's interaction with the interface, and to collect information about potential usablity problems and user preferences at first hand. However, for successful user involvement, detailled planning is necessary, often involving organizational or financial effort. The application of statistical analysis yields to more comparable evaluation results and might in some cases allow for deriving some generalizations or prognoses concerning the evaluated interface. Statistical analyisis is most frequently combined with the strict controlled experiment technique, but it's also applicable for analyzing data, gathered through more informal methods as questionnaires or standard usability tests. In the following, the user-based evaluation techniques (listed in the upper left box of Figure 1) are described in alphabetical order.

1. Controlled Experiment

Sometimes also referred to as *classic experiment*, the *controlled experiment* is a powerful technique to evaluate specific design aspects or even the overall interface design (Dix [35, p. 329]). Originally derived from the field of psychology, this method is believed to yield more objective evaluation results than other evaluation techniques. Basically, a strictly specified kind of user study is conducted, supplemented by some form of performance measurement. One or more hypotheses, that are to be tested, are selected along with a number of dependent and independent variables, the latter of which are varied throughout the experiment. Further, the measurements are defined that are to be collected—for example, task performance time. After users have completed the experimental task(s), the resulting data is examined by means of statistical analyis and the hypotheses are checked. The complexity of the experimental design and statistical analysis constitute the main disadvantages of this technique. A broader introduction how to apply the controlled experiment technique for usability evaluations is given in Dix [35, pp. 329 ff.].

2. Focus Group

The *focus group evaluation* is basically a group discussion with representatives of the target user group. A moderator brings together about six to nine users (Nielsen [106, p. 214]) for a group session, and starts and guides a discussion on interface features. These are are mostly based on

representations such as paper & pencil drawings, storyboards, or even fully functional protoypes. The somewhat informal technique can be applied at any stage in the development process to assess the users' perceptions and needs. The advantages of focus group evaluations are, that they can reveal users' spontaneous reactions towards an interface, as well as their needs and feelings. Altough, it has to be noted that opinions and responses may easily be influenced by the group activity and thus not represent the original position of each user. Furthermore, the technique cannot examine any actual behaviour of the users, but only their perception of what they think they would do or would like to do—which in many cases can differ widely from reality. The potential use and misuse of the focus group technique is further discussed by Nielsen [110].

3. Physiological Monitoring

Compared to other techniques, that mostly rely on the judgements of either users or evaluators, *physiological monitoring* is believed to provide more objective evaluation results through accurate measurements. Although the application of monitoring methods for usability evaluation still is an emerging approach, it is yet quite promising. A successful application of these techniques might yield valuable insight of user's actions, behaviour, and emotional state.

According to Dix [35, pp. 352 ff.], eye-tracking and physiological measurement are the two techniques in this field, that receive most attention today. In *eye-tracking*, the movements of the users' eye activities are recorded. This includes the identification of screen areas, that were longer viewed than other areas and denoting the duration of the fixation. Furthermore, the eye-movement paths are recorded to reveal possible patterns. An analysis of these measurements can indicate potential usability problems; to date however, still more research is required in order to better determine the relation of measured data and possible conclusions for usability.

Physiological measurement on the other hand can provide a means to determine users' emotional responses to an interface, as emotional reactions are closely tied to physiological changes. In measuring such physiological reactions it can be analyzed, which interface components cause stress for the users and which parts rather promote relaxed and enjoying usage. Possible measurments are heart activity, sweat glands activity or electrical activity in muscles or in the brain. Here also more research is required to determine how possible measures and potential usability problems are related.

4. Post-Task Walkthrough

The *post-task walkthrough*—or *retrospective testing*, for example, Nielsen [106, p. 199]—is a technique to compensate for the main shortcoming of observational data: their lack of interpretation. Most user studies or observations reveal the apparent actions of the user, yet do not provide information on the user's thoughts and perception. Even if *thinking aloud* (see also the section about the *user study* below) is applied, the user-specific information might still be insufficient, because users either might have felt uncomfortable thinking aloud—hindering them to use the technique extensively—or because they just mainly concentrated on the task at hand and therefore forgot to do talk during the study. In such cases, a post-task walkthrough can help gaining additional information. Therefore, a transcript—for example, notes from the observer, audio or video recordings, screen activity protocols—is replayed or shown to the participant, who is then asked to comment on his actions or answer specific questions of the evaluator. This permits the participant to concentrate on talking about his perception and experiences during the study. However, it is important to consider whether to use the post-task walkthrough right after the usability test or sometime later. A immediate application after the usability testing has the advantage, that the user still remembers the procedure in general as well as—probably important—details. Conducting the walkthrough later enables the analyst to develop specific questions and focus on interesting incidents (Dix [35, p. 347]). This variant however bears the risk, that the users might either not be accessible anymore, or might not be able to recall the details of the evaluation session to provide any valuable feedback.

5. Query Techniques

Query techniques require users to answer questions asked either by an evaluator or in various forms of questionnaires. That way, additional information from users can be collected, which makes query techniques a powerful supplement to other techniques. Questioning can also be conducted online (see also the section about *remote usability testing*), with the advantage of reaching potentially more diverse and a greater number of users at lower costs.

There exist two types of querying: the *interview* and the *questionnaire*. Interviews are basically more or less guided discussions with the user. Their main advantage is, according to Dix [35, p. 348], the flexibility they provide. It is possible to adapt the questions depending on each user and to discuss important or interesting issues just as they arise during the interview. Cooper [27, pp. 58–68] provides further information on variations of the interview technique.

In contrast, questionnaires are less flexible, as the questions have to be assembled in advance of the session. This also results in the same question set for each user, which is helpful for comparison and for analytical purposes. According to Dix [35, pp. 348 ff.], there exist several types of questions: *general questions*, covering the background of the user—for example, demographic questions or questions about prior knowledge and experiences; *open-ended* questions, that enable the user to provide his opinion in form of writing free-text; *scalar questions*, asking the user to rate a statement on a sepcific scale; *multi-choice questions*, where the user has to choose one or more given response options; and finally *ranked questions*, that ask the user to rank given response options. Questionnaires consisting of rather few and mostly general questions are sometimes also referred to as *surveys*. Various predefined questionnaires already exist for different purposes and different application fields. Examples are the *QUIS* (Questionnaire for User Interface Satisfaction [21]), which is explained in detail by Shneiderman & Plaisant [139, pp. 152 ff.] or the *IsoMetrics* [47]. Some additional information on query techniques are provided by Shneiderman & Plaisant [139, pp. 150–162], Dix et al. [35, pp. 348–351], and Dumas & Redish [39, pp. 546–548].

6. Remote Usability Testing

Remote usability testing (or *remote usability evaluation*), is characterized by the separation of usability evaluators and test users in time and/or space (Paternò [123]). A general advantage of this technique are the lower costs and less organizational effort (Shneiderman & Plaisant [139, p. 149]) when compared to traditional evaluation methods, as users do not have to be brought and tested in a special facility. Therefore, often also more potential participants for the evaluation are available.

According to the definition above, also a simple user study in terms of a field study with users self-reporting counts as remote testing. Hartson & Castillo [52] have proposed such a procedure. In the study they found, that after some minimal training effort users are able to identify, report and severity-rate their own critical incidents occuring while using the interface. For the test evaluation, the authors augmented the interface with a button, users could click, once an incident happened. This opened a standardized incident reporting form, which was, once filled out, sent to the evaluators via the internet, along with an automatically captured video sequence of the users' actions around the time of the incident. As this kind of evaluation takes place in the familiar workplace of the users, evaluation results probably show a more practical orientation.

The rising usage of the internet as a communication medium as well as technological improvements have lead to the development of additional remote testing procedures. According to Hartson & Castillo [52] several web-related types of remote testing can be distinguished. The first one is the *remote questioning*, where interviews or questionnaires simply are deployed and collected through the internet or other web-based technologies as for example email. A variant of this approach is to directly augment the interface to evaluate, so that appropriate questions are displayed right during usage. A shortcoming so far is that the evaluator has no possibility to monitor the test session or to interact with the user in case of problems or questions. This can be compensated by *live- or collaborative remote evaluation*. Here, additional means as, for example, telephone- or

videoconferencing tools are used to enable the evaluator to communicate with the participant and observe his actions. Also recording a user's performance in form of vidotaping is possible that way. Stone et al. [143, pp. 475–477] also propose the usage of special browser logging tools, which falls into the category of *instrumented or automated data collection*. Here, special tools for interaction logging and subsequent automatic calculation of selected usability metrics are utilized to collect and analyze user data.

7. User Study

The *user study* technique often is also referred to as *usability testing*. Basically one must differentiate between field studies (testing is conducted in the users' natural working environment) and laboratory studies (users are asked to perform the test in a testing facility under controlled conditions). A comprehensive overview of usability testing and various additions are presented by Rubin & Chisnell [132] in their "Handbook of Usability Testing".

The *standard form* of the user study basically consists of users performing representative tasks and observing them while they interact with the interface. Therefore, the technique is also referred to as *user observations* sometimes, e.g. Stone et al. [143, pp. 20 ff.]. Nielsen [106, pp. 207–208] on the other hand defines the *observation technique* simply as a user observation in their natural workspace, that is, as a form of field study. The appropriate numbers of test users is controversely discussed. Nielsen [106, pp. 105–108], for example, suggests iterative usability testing with a rather small number of participants each time, as also a small number of test users is sufficient to detect the most serious problems and enables a quick revision of the interface. Moreover he recommends to perform at least two iterations (that is, to evaluate three times in total), as sometimes new problems appear after having revised an interface for the first time. Testing with a small number of participants—between three and six—is also referred to as part of the so-called *discount usability engineering approach*, also introduced by Nielsen [106, pp. 17 ff.]. In contrast, critics point out, that more complex systems can only be thoroughly tested with a broader set of users, for example, Lewis [88]. This conforms to a suggestion of Tullis & Albert [152, p. 119] to recruit 5 participants per siginificantly different class of users, as complex interfaces often are used by more than one class of users.

One factor that influences the usefulness of user studies is the applied *protocolling* method. The standard protocol technique almost always applied is the *paper & pen technique*. Here, the evaluator takes notes on the observed user actions and remarks during the study. Other forms are *audio/video/computer screen recording or logfile analysis* (the users' actions are recorded automatically by using technical media), and the *diary technique* (mostly used in combination with field studies; the users themselves protocol their action in given intervals). The latter is, according to Cooper [27], especially apt for evaluating a design for intermediate or experienced users. A broad review of the diary technique is presented by Kuniavsky [85, pp. 369 ff.].

In a standard user study the participants are characteristically tested one by one and mostly no interaction of the evaluators takes place. As simply watching the users is often insufficient (Dix [35, p. 343]), one addition to the technique is to ask the users to *think aloud*. That means, users are encouraged to communicate, why they perform an action, what they like or dislike about the interface and the tasks, or where they face problems right as they perform the task. This can yield valuable insights about the users' perception of interacting with the interface. The so-called *co-discovery method*—also sometimes referred to as *constructive interaction* or *co-discovery learning*—is a variation of the user study and thinking aloud. Here, participants are not individually tested, but in pairs of two. The main advantage is a more natural test situation for the users (Nielsen [106, p. 198]), as people are used to talking to each other when solving a problem together in contrast to talking to themselves or the audio recorder, when asked to think aloud in a standard study. Thus, users are likely to communicate more freely, which can in turn reveal more usability problems. The co-discovery method is also described by Dumas & Redish [40, p. 31] as well as by Rubin & Chisnell [132, pp. 306–307].

Downey [36] also proposes the technique of *group usability testing* in a recent research article.

Here, several participants perform given tasks individually, but simultaneously, which is supervised by one or more experts that conduct the evaluation. The author considers the technique especially appropriate when many users are available but only limited time or budget. It has to be remarked, that thinking aloud is not an appropriate means to apply here, due to the group session. Therefore more than one evaluator should monitor the testing session so that users' actions and reactions can be observed sufficiently.

Another variation on the standard user study is the *coaching method* (for example, Sarodnick & Brau [133, p. 164], Nielsen [106, pp. 199-200]). Here, the strict separation of task performance and observation is abolished, and interaction of the evaluator and the participant is explicitly desired. Users are encouraged to ask system-related questions whenever they need to, and the evaluator answer them to the best of his abilities. Alternatively it is also possible to nominate an experienced user as the coach (Nielsen, [106, p. 199]). The technique of *active intervention*, described for example by Dumas & Redish [40, pp. 31–32] is quite similar to the coaching method, as both have in common, that the evaluator is free to interact with the participant whenever needed. In contrast to the coaching method, the evaluator does not explicitly explain interface features or procedures to the participant.

The user study technique not only offers the possibility to observe users and gain their individual feedback. It can also be used to collect precise, quantitative usability-related performance measures—often referred to as *usability metrics*—as, for example, *task duration* or *error rate*. The measurement of usability metrics sometimes is also referred to as *performance measurement*. An introduction to this technique is provided by Nielsen by his 'Alertbox' from January 21st, 2001 on his website [105]. According to Stone et al. [143, p. 441], the measurement of usability metrics serves two purposes: on the one hand it permits comparison between different versions of an interface, on the other hand an interface can be evaluated against a set of predefined performance measurments—for example, *maximum task time*. The main requirement of metric based evaluation is, that evaluation materials (participants' introduction, tasks, questions) have to be exactly the same for each user, so as to actually receive comparable results. An extensive list of metrics is provided by Nielsen his book "Usability Engineering" [106, pp. 194–195]. Other possible usability metrics are presented by Constantine & Lockwood [25, pp. 454 ff.], Bevan & Macleod [10], and Rubin & Chisnell [132, pp. 166, 249 ff.]. We assembled a listing of the most popular metrics in Appendix B. Apart from metrics that measure only one aspect at a time, Sauro & Kindlund [134] have developed an approach to combine several usability metrics into a single score, the *SUM (Single Usability Metric)*. Tullis & Albert [152] provide detailed information on various techniques for measuring and analyzing different metrics in their book "Measuring the User Experience".

2.2 Expert Evaluation

Expert evaluation—sometimes also referred to as *usability inspection*—does not require user participation, which is the main difference to user-based evaluation. The assessment of the system or interface is conducted by one or several experts. For nearly every inspection technique specific recommendations of the appropriate expertise of the evaluators exist. Yet, Nielsen [113] generally recommends evaluators with some expertise in usability guidelines, user testing, and interface design, as this leads to a more effective detection and reporting of usability problems. Basically, the inspectors assess the interface, trying to identify issues, that are likely to cause problems for end users. Compared to user-based methods, these techniques are considered relatively cheap in terms of organizational or financial effort. Also they can easier be performed iteratively and nearly at any stage throughout the development process. However, the main disadvantages are the dependency between the inspector's expertise and inspection results as well as the lack of first-hand information from potential users. The latter is especially a problem, as even experienced expert analysts might not be able to estimate correctly how "typical users" will behave. In the following, the techniques presented in the upper right box of Figure 1 are listed and described in alphabetical order.

1. Consistency Inspection

During a *consistency inspection*, an interface is examined in terms of its consistent design. One variant examines the *internal consistency* of interfaces. As Nielsen [106, p. 132] puts it, the same kind of information should always be presented in the same way including not only formatting issues as consistent coloring or the choice of fonts, but also the location of the provided information within an interface. Reviewing such issues can be eased through the usage of guidelines or checklists, which describe the desired properties. Nielsen further remarks that task and functionality structure of a system have to be examined, too, as it is also essential, that no inconsistencies between users' task expectations and a system's task representations occur.

Another variant on the technique described, for example, by Shneiderman & Plaisant [139, p. 142] compares several products within the product line of a company to ensure a consistent design of standard features and thus enable users to identify with products of the same company, and also increase learnability and ease of use. A recent adaption of this technique is described by Chen et al. [20]. For the inspection, they propose two methods: the method of *paired comparison* and the method of *in-complete matching*. The former consists of creating pairs of the interfaces which are then ranked by users in terms of specified properties. The latter aims at assessing the identification of products. Here, users have to assign given product names to the given interfaces, which can reveal whether a product line or company brand can easily be identified. Consistency inspection can be supported by *birds-eye viewing*: laying full sets of printed screenshots on the floor or pinning them to a wall to provide the evaluator with an overview and to ease comparisons.

2. Feature Inspection

Another inspection technique applicable for usability evaluation is the *feature inspection*. As Nielsen [108, p. 6] describes, feature inspections focus on the functions delivered in a software system. Therefore, the intended user tasks are examined first, and all features (that is, system functionalities) required for fulfilling a certain task, are listed, followed by a an examination of the sequences of features. The evaluator aims at identifying whether those sequences are too long, or contain cumbersome steps a user would not likely try, or steps that require extensive knowledege or experience. Therefore, this technique is quite similar to the *walkthrough technique* as presented below, as they both aim at identifying steps in task sequences that might not be natural for a user to try or only difficult to assess. The difference between the techniques is that the feature inspection emphasizes the functionality and the availability of a system's features, that is, it is examined, whether the functionalities meet the user's needs for task fulfillment. In contrast to that, the walkthrough has the major goal to examine the user's problem-solving process while trying to perform a task and to investigate the understandability and moreover the learnability of the interface.

3. Formal Usability Inspection

The *formal usability inspection* is summarized by Nielsen [109] as a "procedure with strictly defined roles to combine heuristic evaluation and a simplified form of cognitive walkthroughs". Kahn & Prail [75, p. 141] specify the technique more strictly, describing it as a "formal process for detecting and descibing defects, clearly defined participants' responsibilities, and a six-step logistical framework". A main characteristic though is the participation of several inspectors—Kahn & Prail recommend four to eight inspectors that ideally possess different levels of expertise. The role of each team member as well as the complete procedure are strictly defined, and described in detail by Kahn & Prail [75]. In summary, the evaluators try to identify certain user profiles (created in a preliminary target user analysis) and step through representative task scenarios. In addition to this walkthrough-like procedure, inspectors apply given heuristics while stepping through the tasks, and afterwards describe the usability defects found, again in a strictly defined manner to enable clear communication of the issues. The authors point out, that any appropriate list of

heuristics, and even more than one set, can potentially be applied. Yet they recommend Nielsen's set of 10 heuristics [106, 108] (see Appendix A.1) as it is both brief and comprehensive. As more they assembled an own, more elaborate set of heuristics [75, p. 150] from Nielsen's 10 heuristics, Shneiderman's rules for interface design [139, pp. 74-75] (see Appendix A.7), and other relevant literature not further named.

4. Guideline-based Evaluation

A *guideline-based evaluation*, sometimes also referred to as *guideline/standards review* or *guideline/standards inspection*, is an interface inspection in context of either organizational, governmental or scientifically established standards or guidelines concerning usability and design. Checklists are prepared before the evaluation on the basis of the chosen guidelines, and are often used to simplify the evaluation process.

To conduct the evaluation, an inspector explores the interface and checks for its conformity with either the guidelines or the concrete checklists. Already established guidelines are often sets of a large number of items and often contain detailed directives on how to design particular interface components. Therefore, this technique can be used by usability professionals as well as by non-expert software developers or interface designers. Due to the huge number of items, the process of selecting the relevant items or preparing appropriate checklists may be time consuming. The same holds according to Shneiderman & Plaisant [139, p. 142] for the evaluation itself, especially when complex interfaces are inspected. The problems that can be detected easily through a guideline-based evaluation mostly include design, layout, or consistency flaws; more severe problems, as, for example, navigational or structural misdesigns, may be missed. This is the reason why this technique is best used in combination with further usability evaluation techniques, that focus on usability-related issues other than on design.

To date, a huge amount of guidelines for varying purposes is available. Some known guidelines include the ones created by *Smith & Mosier* [141] and the more recent *ISO 9241 standards*, provided by the International Organization for Standardization [66]. Nielsen [106, p. 94] also shortly introduces some more sets of guidelines. For an ISO 9241-based evaluation, there also exists an extensive handbook, the *DATech-Leitfaden Usability* [28], released by the german accreditation body in 2008. Moreover, many books on interface- and webdesign generally provide helpful guidelines or at least serve as a basis for developing some own. Among these, the most remarkable ones are "About Face" (Cooper [27]), "GUI Bloopers" (Johnson [72]) or "The Essential Guide to User Interface Design" (Galitz [46]).

5. Heuristic Evaluation

Heuristic evaluation is similar to the guideline-based evaluation in that both techniques apply a set of guidelines as the basis for the inspection. Yet at the same time those guidelines are the main difference between both techniques. Whereas the former approach makes use of extensive sets of quite detailed guidelines, those used in a heuristic evaluation (called heuristics, instead of guidelines) are a lot more general. As more, the number of heuristics is relatively small, about 10 to 20 heuristics is the common practice.

An evaluator assesses the interface's conformance with the chosen heuristics to conduct a heuristic evaluation. The evaluator explores the interface several times, and compares interface and dialogue elements to the list of heuristics. Due to the frequently abstract formulation of the heuristics a considerable amount of expertise is required from the inspector for a successful evaluation. Shneiderman & Plaisant [139, p. 142] for example note, that the results of heuristic evaluations can widely vary between such evaluators, that are familiar with the heuristics or at least able to interpret and apply them, and such evaluators, that do not possess these abilities. The lack of knowledge of rather inexperienced evaluators can be compensated to some extent, if more detailed checklists are created on basis of the general heuristics. Moreover, the *participatory heuristic evaluation* technique was developed (see section 2.3) that even enables non-expert users

to take part in a heuristic evaluation.

Although it is possible that a single evaluator—typically the interface developer or designer himself—conducts the heuristic evaluation, Nielsen [106, p. 156] strongly recommends ideally three to five evaluators for the most cases. He points out, that different inspectors not only find a greater overall number of usability problems, but also distinct ones, but emphasizes that evaluators have to inspect the interface each one on their own to achieve this desired effect. As the technique potentially finds major as well as minor usability problems, occurring in terms of violations of one or more heuristics, Nielsen also suggests a severity rating of the detected problems. Moreover, he points out that heuristic evaluation may in some cases miss specific usability problems, for example, when non-domain experts are inspecting a highly domain-dependent interface. He therefore recommends to additionally conduct a user-based evaluation, such as the user study.

To date, several sets of heuristics are known. The probably best known and widely applied are the 10 heuristics of Nielsen [108]. Those were complemented by three more rules in 1995 by Muller et al. [102]. Muller et al. also developed the approach of *participatory heuristic evaluation* (section 2.3), in the course of which they adapted Nielsen's heuristics, to compose a set of 15 heuristics on their own. Another quite well-known set, Shneiderman's so-called eight golden rules [139] consists of just eight heuristics. A more recent article of Kamper [76] describes his efforts to develop a set of simple, unified and interdisciplinarily applicable heuristics - resulting in a list of 18 heuristics, based on Nielsen's original 10, but adapted to serve his intended purpose. The complete listings of the most popular sets of heuristics along with short explanations and comparisons between the sets are provided in Appendix A. Apart from those sets of heuristics that were specifically developed and intented for heuristic evaluaion, also many sets of basic interface design guidelines exist. These are in some cases also quite generally worded and may thus be used as the basis for a heuristic evaluation, too. Examples are the design guidelines of Donald A. Norman [117] and the *8 golden rules of interface design* of Ben Shneiderman [139, pp. 74-75]. These are, among others, also listed in Appendix A.

6. Model-based Evaluation

As their name suggests, *model-based evaluation techniques* use models of interfaces as the basis for the evaluation. The goal is, to predict mostly quantitative measures of an interface—for example, task duration—by simulating the users' behaviour. The basic technique consists of 4 steps: describe the interface design in detail, create a model of representative users and their task performance, predict chosen measures by simulating the model, and finally revise or chose the design depending on the prediction. Such a simulation can take place at early stages in the developement process and thus valuable usability results can be collected without even implementing a prototype. However, it can be challenging to correctly set up and fine-tune such a model and, even when done, it still might not be a complete or perfected mapping of the actual interface.

Several models applicable for usability evaluation already exist. Probably best-known is the family of *GOMS models*. The GOMS describes **G**oals (the goals, a user can accomplish with the system), **O**perators (basic ations, such as mouse clicks or finding an icon), **M**ethods (sequences of operators, required to accomplish a certain goal) and **S**election rules (which describe, which method is required to accomplish a certain goal). There exist several variants of GOMS models: the *Keystroke-Level Model* (KSLM), the *Card, Morn and Newell GOMS* (CMN-GOMS), which is considered the basic GOMS, the *Natural GOMS Language* (NGOMSL), that builds on the CMN-GOMS providing a natural language description for the methods, and finally the *Cognitive-Perceptual-Motor GOMS* (CPM-GOMS). The CMN-GOMS—developed on the basis of the KSLM—are originally presented by Card et al. [18, 19]. An extensive description of the different GOMS models along with suggestions on the proper contexts of usage, is provided by John & Kieras [70, 71].

Apart from the GOMS models, there exist various other forms of cognitive models that can be adapted for interface evaluation purposes. Examples are the *task networks* and the *cooperative architecture model* as summarized by Kieras [81]. As he explains, the former basically model task

performance in terms of a network of processes, whereas the latter are composed of cognitive and motor components as well as components of the human perception. A more detailed survey on cognitive architecture systems in general, and on former as well as more recent architectures is provided by Byrne [16].

In his book "Model-Based Design and Evauation of Interactive Applications", Paternò [123] also presents an extensive introduction and overview to model-based design and evaluation, thereby focussing on task analysis and thereupon based models. He also introduces an own method for usability evaluation of interfaces: the *RemUSINE* method combines task-models, remote evaluation and logfile analysis into one evaluation approach. Dix [35, pp. 420 ff.] provides an overview on several model-based techniques, too. Apart from GOMS and KLM models he also summarizes the *cognitive complexity theory* approach or *linguistic models* including the BNF- (Backus-Naur-Form) or the task-action grammar approach.

7. Using Previous Studies

Another approach for the evaluation of interfaces is the analysis and usage of results from *previous studies*. Considering usability evaluation there already exist a multitude of experimental results and empirical evidence from the fields of experimental psychology and human-computer interaction (Dix [35, pp. 326–327]). Under certain circumstances some of these previously gained results can be used as evidence to support certain design aspects of an interface or on the other hand to rebut them.

The difficulty with this technique is the relatively high level of expertise that is required of the evaluator, necessary for selecting adequate research studies and results that closely match the given context. It has to be considered, whether all basic conditions—such as experimental design, the choice of participants, assumptions made, or analyses applied—are matching, so that previous results can be applied to the actual context. Therefore, when using this technique, it is also highly advisable, to not only consider, but carefully denote similarities and especially differences—for example, when not all conditions are matching appropriatly—between previous research and the actual work.

8. Expert Walkthrough

To date there exist several variants of the *walkthrough technique*. Probably best-known and also the basis for other variants, is the *cognitive walkthrough*. Here, the evaluator steps (mostly mentally) through several action sequences of the interface, that are necessary, to perform some defined tasks. The challenge and key requirement of the walkthrough technique is, that the evaluator has to put himself into the position of a potential target user, which demands both cognitive skills and a good knowledge and understanding of the users and their goals. As the evaluator steps through the actions, he considers after each step, whether it was appropriate for achieving the overall goal, and whether the user can figure out, which step to take next at this point of action.

For the conduction of a walkthrough, it is essential, that a detailed task description, broken down to a sequence of single actions, is available. Shneiderman [139, p. 142] recommends to use the most frequent tasks as a starting point, but to also include rare critical tasks into the walkthrough. Moreover, a prototype description of the system is needed—this is not necessarily required to be complete, but those parts, addressed throughout the tasks, should be specified in detail.

The cognitive walkthrough mainly focusses on the ease of learning of the interface, but as the attributes ease of use and functionality are highly correlated to learnability, they are indirectly addressed, too. As Wharton et al. [158] describe, the technique can be conducted both by a group of evaluators as well as by a single evaluator. They further recommend to use the technique in combination with other evaluation approaches, to minimize the risk of one-sided, learnability-focused results. A comprehensive description of the whole procedure is provided by Wharton et al. [158].

A variation on this technique is the *pluralistic walkthrough*, further described in section 2.3.

Another variant, the *heuristic walkthrough* is proposed by Sears [137], combining both heuristic evaluation and cognitive walkthrough. Basically, the evaluator is given a prioritised list of user tasks. In a first phase, the expert uses a set of thought-provoking questions to assess the system like performing a cognitive walkthrough. Afterwards, a free-form exploration of the system is performed, using both the list of questions as well as heuristic evaluation. Sears [137] provides a more detailed description of this method.

The walkthrough techniques are all applicable in early stages of a product's design, without the need for a fully functional prototype. This makes them a rather cheap yet valuable utility for early usability evaluations.

2.3 Hybrid Approaches

This section introduces usability evaluation techniques that cannot be put clearly to the former two categories. In the cases of *collaborative inspection*, *participatory heuristic evaluation*, and *pluralistic walkthrough* the technique consists both of expert- and user-based evaluation parts, which makes them true, hybrid approaches. *Competitve analysis* and *usability evaluation of websites* on the other hand are to be seen as possible additions to standard usability evaluation techniques. As the latter both can be used with user-based as well as with expert evaluation techniques, they cannot be clearly integrated either as a user-based or as a expert evaluation technique.

1. Competitive / Comparative Analysis

In general, the main characteristic of a *competitive or comparative analysis* is that more than one evaluation is performed and the results are compared afterwards. More precisely, two variations of the technique have to be distinguished. The first one—oftem referred to as *competitive analysis*—mostly consists of evaluating existing interfaces of competing products. A product similar to the one to be evaluated is regarded and treated as an own prototype to conduct a first evaluation with. Clear advantages are that the time and the costs for developing an own prototype can be saved; moreover, if the competing system is already fully developed the testing is more realistic than with a prototype (Nielsen [106, pp. 78–79]). Design flaws and usability problems detected can provide valuable insight on how to design—or not to design—the own interface. Moreover, a simultaneous evaluation of both the own and the competing interface can be conducted to directly compare the two designs against each other.

The second variant, mostly referred to as *comparative evaluation* is characterized as carrying out simultaneous evaluations of the same interface with multiple evaluation teams, applying a single or multiple evaluation techniques (Koutsabasis [83]). The main advantage is, that multiple teams of evaluators are likely to detect varying usability problems, even if all teams apply the same evaluation technique. If each team additionally uses different evaluation methods, then the chances of finding a larger number and variety of problems even increases.

2. Collaborative Usability Inspections

Larry L. Constantine [25, pp. 401 ff.] presents another hybrid approach, the *collaborative usability inspection*, in his book "Software for Use". A collaborative usability inspection is carried out by a team of evaluators with defined roles for each team member. Constantine suggests 6 to 12 team members based on the results of his own research. The inspection team consists not only of usability experts, but also of software developers and potential end users. Similar to a pluralistic walkthrough, the team inspects the interface in a group session, but with the difference that here the team openly communicates and collaborates throughout the evaluation session. In a pluralistic walkthrough session, in contrast, each evaluator first works on his own and the results are compared and discussed after each task.

Constantine proposes two phases of a collaborative inspection: the *interactive inspection* and the *static inspection*. In the first phase, the system is either actually used, or its usage is simulated. Here, representative tasks and scenarios—that have to be prepared in advance—should

be addressed. After each task, the evaluators should comment on the interface and/or the task. During the second phase, the inspection team reviews the interaction contexts one after another. Constantine suggests, that at least every interface composite—for example, screens, dialog boxes, menus—should be visited once. Thereby as many fine details as possible should be assessed, as for example icons, labelling, or messages. In his book [25], Constantine provides further information on the roles of the team members and the whole inspection process.

3. Participatory / Cooperative Heuristic Evaluation

Both *participatory* and *cooperative heuristic evaluation* resemble the standard heuristic evaluation technique as described in section 2.2. The basic procedure—assessing the interface with the help of a set of heuristics—is unchanged, but both users and experts are incorporated as evaluators into the evaluation.

The first variant, the *participatory heuristic evaluation*, is generally guided through task lists or scenarios. Users are asked to participate the evaluation as work-domain experts and equal evaluators next to the HCI or software experts, for example, Muller et al. [101]. The main benefit is that users are able to complement the knowledge of traditional experts, that is sometimes more theoretical or abstract. This is mostly due to the fact, that users often possess a more practical knowledge about the actual usage and requirements of the interface or software. Muller et al. also adapted Nielsen's original set of heuristics. This was done in two ways: first the authors added some process-oriented heuristics for a more balanced evaluation, as they found the original heuristics rather primarily product-oriented. This resulted in a revised set of 15 heuristics. Moreover, the wording of the heuristics was refined to ensure, that also evaluators without specialized software or usability knowledge would correctly understand the heuristics and be able to apply them. Take, for example, the fourth of Nielsen's 10 heuristics (refined version of 1994)—*Consistency and Standards*—that says:

> "Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions." (heuristic #4, Nielsen [108, p. 30])

Though the name of the heuristic–*Consistency and Standards*—has been retained, the description of the heuristic has been revised by Muller et al. [101] into:

> "Each word, phrase, or image in the design is used consistently, with a single meaning. Each interface object or computer operation is always referred to using the same consistent word, phares, or image. Follow the conventions of the delivery system or platform." (heuristic #6, Muller et al. [101, p. 16])

Whereas the wording of Nielsen's version is kept rather concise, the heuristic of Muller et al. provides a somewhat more elaborate explanation, that might be better understood by non-expert evaluators. The complete listings of both sets of heuristics are provided in the appendix: Nielsen's 10 heuristics in Appendix A.1, the revised set of Muller et al. in Appendix A.2.

The second variant of the heuristic evaluation technique is the *cooperative heuristic evaluation*, proposed by Sarodnick [133]. Similar to the participatory heuristic evaluation users participate in the evaluation session, but with the difference that each evaluating team consists only of a pair of evaluators: one expert-evaluator and one user. In advance to the evaluation, task szenarios are developed and the experts are taught the correct usage of the system. During the course of the actual evaluation, the expert performs the tasks while the user is also attending the session. The user is encouraged to comment on the expert's actions whereas the expert should ask the user comprehensive questions, explicitly concerning the underlying, real work sequences. Therefore it is crucial that the user is able to articulate the underlying, real work processes in a clear and structured way. In having the user comment and explain the work sequences, the expert is supported in adopting the point of view of a future actual user of the system. The whole procedure is intended to facilitate a heuristic evaluation exceeding a pure inspection of the interface despite the complexity of the work domain. Apart from the named skills of the user, also advanced communication skills as well as an ability to quickly adapt a user's point of view are required of the expert evaluators

to perform a cooperative heuristic evaluation successfully.

4. Pluralistic Walkthrough

The *pluralistic walkthrough* was introduced by Bias [12]. Its main difference in comparison to the cognitive walkthrough is the participation of more than one evaluators of different types: representatives of the target user group, developers and usability experts. Basically, each participant then receives a set of interface print-outs and task descriptions. Then participants are asked to write down what actions they would perform to accomplish the first given task on basis of the corresponding printout. Thereby it is essential, that participants provide a description as detailed as possible. After each participant has completed his description, the "right" solution (the solution the developer intended) is presented and participants' solutions are discussed with the expert. Bias also suggests that the users should fill out a questionnaire after each task and also after the complete walkthrough session has ended. The advantage of this technique lies in the involvement of both experts and users. That way, both expert and end-user knowledge can be collected at first hand. A comprehensive presentation of the pluralistic walkthrough is presented by Bias [12].

5. Usability Evaluation of Websites

The *usability evaluation of websites* is an adaptation of evaluation techniques for the specific context of examining intranet sites or websites, thus most of the techniques described here are based on those presented in sections 2.1 and 2.2. Still we discuss usability evaluation of websites separately, mainly because websites constitute a special category of applications. They are utilized—compared to stand-alone applications—by a lot more users which also leads to a larger amount of non-expert users. This in turn as well as specific properties of websites concerning their design and usage raises the need for an even more careful evaluation and tailored modifcations of the basic evaluation techniques. One of those properties is that websites are mainly focused at presenting information to the user. This implies the need for specific architectures for navigation and information, special ways of presenting site contents or of implementing information searching-related features. Another problem with websites, as Krug [84] remarks, is that they are rather superficially scanned by users than read through, which in turn also implicates special design desicions. In the following, we describe approaches to modify evaluation techniques for website evaluation, as well as some recent developments on new techniques.

If specifically tailored guidelines or heuristics are chosen as a basis, both *heuristic evaluation* and *guideline-based evaluation* are applicable to websites. Appropriate heuristics are for example presented by Levi & Conrad [87]. Borges et al. [14], Spool [142], as well as Nielsen [116] have proposed and refined specific guidelines for website design, which can serve as a basis for evaluation, too. In a recent article, Bevan & Spinhof [11] present their research on developing a new set of guidelines and a checklist for evaluating web usability. As a basis, they use the draft International Standard ISO/DIS 9241-151 as well as the revised guidelines from the U.S. Department of Health and Human Services [153].

Another technique that can be easily adopted for the usability evaluation of webistes is the *usability testing technique*. Both Krug [84, pp. 139-181] and Spool [142, pp. 143-153] provide some further information on this topic. In general, the basic testing technique, including augmentations as thinking aloud, can be applied as explained above, but the types of tasks, that users have to solve, should be adjusted. Thus, tasks should rather aim at examining navigation and information-seeking issues, as these are the main activities, users of websites are likely to perform.

Moreover, it is advisable also for the context of website evaluation, to conduct some *user questioning*. There already exist predefined questionnaires for website evaluation, as for example the *MIT Usability Guidelines* [65] and the *Website Analysis and MeasureMent Inventory - WAMMI*, as proposed by the Human Factors Research Group Ireland [149].

Recently, another technique has been proposed for website evaluation: the usage of *logfiles*. In their recent article, Fujioka et al. [44] describe their research on using mouse click logs for detecting

usability problems of websites. They aim at identifying unnecessary or missed actions (which they consider cues of usability problems) by comparing actual user interaction logs of mouse clicks with desired click sequences for the task. Moreover, they developed a tool for logging and analyzing the data in terms of their proposed method.

As said, navigation plays an important role for the usability of a website. López et al. [93] recently presented the tool EWEB which enables automatic empirical evaluation of web navigation. Their tool supports several techniques—for example, storing web logs and using questionnaires. The different techniques can either be used separately or in combination by the evaluators. Also selected usability metrics—for example, *success rate* and *task duration*—can be calculated automatically based on the logging of the user's site navigation behaviour. Moreover, the authors provide an overview of currently available tools for capturing various forms of user interaction with websites.

Zhang & Dran [162] presented yet another approach: a two-factor model for assigning web design factors (as e.g. visually attractive layout) into two categories: *hygienic factors*—those factors that ensure essential functionality—and *motivator factors*—factors, that increase users' satisfaction and motivate them to visit the website again. They argue, that hygienic factors are of higher priority than motivator factors and thus should be tidied up first, before being concerned with motivator factors. Their conclusion is, that both designers and evaluators can benefit from their categorization and prioritization of website design factors.

More extensive information related to website usability is provided by Jakob Nielsen, an expert in this field. Both his website [105] and several of his books—for example, [112, 116]—cover many topics related to webdesign and evaluation. A rather compact introduction to the whole topic is presented by Krug [84].

2.4 Further Approaches

The techniques provided in this section are not specifically designed for usability evaluation. In fact, they are well known approaches mostly from general software engineering, that can enhance usability evaluation when used additionally.

The first approaches to mention are *iterative development*, *pilot testing* and *prototyping*. Iterative development is a common practice in software development, but should also considered particularly for usability evaluation, too. If usability evaluation is performed several times during development (say: first with a paper draft, then with a prototype, and in the end with the real product) rather than just once at the end of development, more potential usability problems are likely to be detected and removed. Moreover, Nielsen [106, p. 105–108] remarks, that redesigning after an evaluation can lead to new, different usability problems. He strongly recommends at least two iterations of usability evaluation, and interface refinement. The second approach, pilot testing, should also be adopted for usability evaluation. Nielsen highly recommends running a pilot test on the usability test to ensure, that the evaluation actually will work, that required materials are complete, that the time schedule is planned correctly and adequate tasks or questions are chosen. For pilot testing, Stone et al. [143, p. 503] suggest chosing a participant that can be confidently be tested with and is right available, rather than searching for a perfect representative of the target user group. As the methods' overview already showed, many evaluation techniques can be applied to prototypes. This provides the advantage (e.g. noted by Nielsen [106]) that usability evaluations can be conducted relatively early in the development cycle, right when first prototypes are available.

The *peer reviewing technique* is also quite well-known. Mostly used rather informally, peer reviewing often consists of simply asking a colleague—who doesn't even have to be a domain or HCI expert—to "have a look at it". Yet it is also possible to chose more formal methods as e.g. a heuristic evaluation. Stone et al. [143, p. 537] further explain, that peer reviewing is especially appropriate to be used in early design stages, as it offers the possibility to collect first impressions and suggestions from other people almost without any effort.

Acceptance tests are suggested by Shneiderman & Plaisant [139, pp. 162–163] as another means to be added to user interface evaluation. Therefore, measurable and objective performance goals (as, for example, the maximal acceptable task performance time for a given task) have to be

determined. Given adequate metrics the interface later can be checked in terms of fulfilling the defined criteria. The central goal of this technique is not identifying new usability or design problems, but to confirm that the product meets defined goals.

As several problems concerning the usability of an interface might not occur until it has been used some time the technique of *evaluation during active use* aims at receiving user feedback after the deployment of the product. One way to collect such feedback is to conduct interviews with representative single users or user groups after a certain amount of time. Moreover, Shneiderman & Plaisant [139, pp. 165–166] also suggest online or telephone consulting services, suggestion forms, bug reports or discussion boards, that can all contribute to collecting first-hand information from actual users and their concerns. It is also possible, to ask users, to use the diary technique (see also section 2.1) over a certain time. Finally, some form of data logging might provide valuable insight about the actual usage of a product. Yet this technique might often be applicable only in a constrained way, as most users might not like their actions to be recorded over longer periods of time.

3 Use of Usability Evaluation in Current Works

So far, the precedent section provided an introduction and categorization of the different usability evaluation techniques known to date. The second goal of our research was to investigate which of the presented techniques are actually applied in real-world evaluations of scientific works. Characteristically for typical applied scientific works are a limited budget of time, money, and participants, when compared to works from the industrial field. Workers in the latter field are, in the majority of cases, able to invest an amount of resources into the evaluation of their products. Thus also more complex evaluations under perfected conditions (for example, using a large amount of participants) are possible. In contrast, we were interested, to what extent evaluation methods are applied in the more constrained applied scientific context.

For our study we decided to focus on Ph.D. and MA theses, as they best reflect constrained conditions as described above. Researchers in this context are often bound to get by with limited financial means, a tight time-frame, and few to none participants for user-based evaluations. As more, mostly no external experts on HCI, usability, or interface evaluation are available. So if researchers of theses want to apply expert-based techniques, they mostly have to conduct evaluations themselves. As they are mostly no HCI-experts themselves, this in turn exposes another interesting aspect: the extent, to which the presented techniques are intuitively applicable by non-experts to evaluate their own systems. Finally, Ph.D. and MA theses mostly provide a detailed view on the methods used and experiments taken, delivering us insight into the common practice of usability evaluation. This chapter introduces the theses that we examined during the course of the research. First an overviewing table (Table 5) is presented, along with a description of its parameters and entry values in section 3.1. Afterwards, the findings are described in greater detail in section 3.2.

The goal of the comparison and the presentation of the theses in the chosen order is not to identify any “loosers” or “winners” with respect to the evaluation method. That is, the theses are not considered better or worse if they applied a single method, or various techniques, or if they preferred using one technique over another, as this is dependent on the underlying system and the context and purpose of the evaluation. As presenting an exhaustive overview of all theses published on the topic of ergonomic interface design and usability evaluation lies outside the scope of our survey, we rather aimed at presenting a picture of the actual, current—years 2000 to 2008—usage of usability evaluation methods in the field of computer science. In doing so we focused on applied scientific theses containing a HCI-related term—as, for example, *usability* or *evaluation*—in the title.

3.1 Synoptical Table and Table Entry Types

All examined theses from the field of computer science or strongly related fields—as, for example, bioinformatics—are summarized in the synoptical table, Table 5.

Table 1: Evaluation Techniques

card	(card sorting, see section 2.1)	**peer**	(peer reviewing, see section 2.4)
ex	(controlled experiment, see section 2.1)	**pilot**	(pilot testing, see section 2.4)
focus	(focus group testing, see section 2.1)	**proto**	(prototyping, see section 2.4)
guideline	(guideline-based evaluation, see section 2.2)	**quest**	(questionnaire technique, see section 2.1)
interview	(interview technique, see section 2.1)	**stats**	(statistical analysis, see section 2.1)
heuristic	(heuristic evaluation, see section 2.2)	**study**	(user study, see section 2.1)
iter	(iterative analysis, see section 2.4)	**walk**	(expert walkthrough, see section 2.2)

Table 2: Attributes

a rec	(audio recording)	**log**	(logfile analysis)
cog	(cognitive walkthrough)	**metrics**	(measuring accurate metrics)
comp	(competitive/comparative anaylsis)	**plur**	(pluralistic walkthrough)
diary	(diary technique)	**post**	(after main evaluation)
expl	(explorative testing)	**pre**	(before main evaluation)
eye	(eye tracking)	**p-t**	(post-task walkthrough)
f	(formal/guided interview)	**remote**	(remote evaluation)
f-b	(short informal feedback)	**s-f**	(semi-formal interview)
field	(conducted in users natural workspace)	**s rec**	(screen recording)
i-f	(informal interview/questioning)	**survey**	(survey technique)
lab	(conducted in laboratory)	**t-a**	(thinking aloud)
		v rec	(video recording)

Before actually presenting the table, its parameters and sorting are explained in more detail. Each parameter conforms to a table column. In the following, the parameters along with an explanation are listed in the same order as they appear in the table. In doing so, the shortened form of the parameters—as used in the synoptical table (bold-face terms)—is provided first, the complete terms are given in parentheses. Where appropriate, predefined entry values for the parameters are described, or listed in associated tables. Here again, the bold-face terms are the short forms as used in Table 5, the complete terms are additionally listed in parentheses.

- #: Each thesis was provided with a consecutive numbering.
- **Method**: the evaluation technique applied in the thesis. Table 1 provides a listing of all techniques that were applied for evaluation purposes within the examined theses. Table 2 additionally presents possible attributes (listed in square brackets in the method column in Table 5) that represent additional features of the evaluation techniques. Those features are listed separately here, as several of them can be used in combination with more than one basic technique—for example, *thinking aloud* could be applied in a user study as well as in an experiment. Where approriate, also the type and number of items used—for example, the number of tasks in a user study—are given in parentheses in the method column in Table 5). Possible item types are listed in Table 3.
- **Par** (Participants): the experience level of the participants involved in the evaluation, that is, evaluators or test users. We applied following categories: **nov** (novice, user with no or very little experience with or knowledge about the system to evaluate), **med** (intermediate, user with some experience or knowledge), **exp** (experienced, user with advanced experience or knowledge), **var** (includes users from all of the 3 previously named groups). Mixed specifications, as for example **med/exp**—both people with intermediate or advanced experience within the group of evaluators—are also possible.

Table 3: Item Types

c	(evaluation criteria—for example, heuristics, guidelines, rules)
d	(demographic questions—for example, age, gender, prior experience or knowledge)
q	(questions concerning usability issues)
t	(tasks that have to be performed during the evaluation)

Table 4: Application Types

mob	(mobile software—software for mobile devices such as PDAs or cell phones)
pen	(pen-based interface)
s-a	(stand-alone/desktop application)
term	(terminal software—for example used for information terminals as found in museums)
w app	(web application—browser based application)
w port	(web portal—intranet site or web forum)
w site	(website—single webpage, or website consisting of several pages)

- **# Par** (Number of Participants): the number of persons involved in the evaluation, that is, the number of test users or (expert-)evalautors, depending on the applied technique.
- **Time/min** (Time in Minutes): the average time in minutes, an evaluation session lasted. This is always the time measured from the point of view of the participants, that is, the test users if a user-based evaluation technique is applied, or the expert(s) if an expert-based technique is used.
- **Source**: the literature containing basic principles or theoretical foundations, the researcher of the thesis used for developing the evaluation. An overview of all the relevant sources utilized within the investigated theses is provided in section 3.2.3.
- **Eval Subject** (Subject of the Evaluation): describes, what exactly has been assessed. This was, in most cases, the overall system usability. When specific properties such as, for example, GUI design were particluarly evaluated, these are listed separately.
- **App** (Application): the type of application, that was evaluated. Table 4 presents the different types found within the examined works.
- **T Users** (Target Users): the level of the target users' prior experience with, or knowledge about, the system examined. Here, the same categories were applied as for *Participants* (see above).
- **Thesis**: the title of the thesis examined, the literature reference, and the year of publication. Here, the title is presented in its native language—whenever the native language is other than english a translation of the original title is additionally given in parentheses.
- **Field**: the precise specification of the field, the thesis was published in.
- **Cat** (Category): **Ph.D.** (Ph.D. dissertations, 15 in total) or **MA** (MA theses, 20 in total)

The main sorting criterion of the table entries are the values of column *Method*, that is, the applied evaluation technique(s). First, all theses applying both user-based and expert-based techniques are listed, followed by all theses, applying only user-based techniques, and finally followed by those, applying only expert-based techniques. Each of these three sets of theses is further ordered by the number of different techniques applied, that is, a thesis applying both questionnaire technique and user study (two distinct techniques) is listed before a thesis applying the user study as the single evaluation method. In the cases, where a table entry for the thesis consists of more than one distinct evaluation techniques, the latter are listed in alphabetical order.

Finally it has to be noted, that each technique is only listed once per thesis. In those cases, where the researchers used a method iteratively more than once, the values concerning this technique—for example, the number of users—was calculated as the average value of all applications of the technique.

Table 5: Usability Evaluation Methods in Current Work

#	Method	Par	#Par	T/min	Sources	Eval Subject	App	T Users	Thesis	Field	Cat
USER- AND EXPERT- Based Evaluation											
1	heuristic (10c)	med/exp	Ø15	30–40	Nielsen93, Nielsen94b	joy of use, entertainment	w site	var	A Measure of Fun—Extending the Scope of Usability [159] - 2003	computer science	PhD
	interview [s-f]	med/exp	10	/	/						
	quest [d]	nov/med	10	/	/						
	study [t-a, field, expl]	nov/med	10	/	/						
	walk [plur]	med/exp	/	30–40	/						
	iter	/	/	/	/						
2	focus	exp	6	30–120	/	navigation, acceptance, satisfaction	s-a	var	Redesign von Benutzungsoberflächen durch Mittel der Navigation (Redesigning User Interfaces by the Means of Navigation) [78] - 2003	computer science	MA
	heuristic	exp	1	300	Tognazzini						
	quest (Ø20q)	med/exp	10	/	DATech						
	study [t-a] (8t)	med/exp	10	60–90	/						
	walk [cog]	exp	6	60–120	/						
	iter	/	/	/	/						
3	eye	/	/	/		ergonomics	w site	var	Ergonomische Gestaltung der Webauftritte: Analyse des menschlichen Verhaltens bei der Webnutzung und darauf basierende nutzerspezifische Vorschläge (Designing Ergonomics into Web Presences: Analyzing Human Behaviour while Using the Web, and User-Specific Design Suggestions) [45] - 2004	computer science	PhD
	interview [f] (22c)	var	27	/	ISO 9241						
	quest (Ø14q, 13d)	var	27	/	/						
	study [t-a, v rec, eye] (Ø5t)	var	27	/							
	walk [p-t]	var	27	/	/						
4	guideline (113c)	exp	1	/	NielsenTahir02	usability, navigation, understandability, usage, quality of contents	w port	var	Nutzen und Nutzbarkeit des Felsinformationssystems des DAV - eine Usability Studie (Use and Usability of the Mountain Information System of the DAV— a Usability Study) [59] - 2007	media & communication science	MA
	interview [comp]	med/exp	5	/	/						
	quest [remote, survey] (Ø45q)	var	Ø112	/	/						
	study [v rec, t-a, log] (Ø4t)	med/exp	13	20-30	/						
5	card	nov/med	5	/	/	overall usability	w site	var	Usabilidade no Contexto de Gestores, Desenvolvedores e Usuários do Website da Biblioteca Central da Universidade de Brasília (Usability in the Context of Managers, Developers, and Users of the Central Library at the University of Brazil) [30] - 2006	informa-tion science	MA
	guideline (109c)	exp	4	/	/						
	heuristic (6c)	exp	4	/	Nielsen94b, Dias01, Nielsen00, ErgoList, BastienScapin93						
	study [t-a, v rec, a rec, s rec] (10t)	nov/med	21	/	/						
6	heuristic (14c)	exp	1	/	Nielsen94b, Norman88, Shneiderman04, Wroblewski01	usability, GUI design	w port	var	Evaluation of the User Interface of a Web Application Platform [157] - 2006	computer science	MA
	interview [f]	med/exp	21	/	/						
	quest (41q)	var	40	/	/						
	study (14t)	var	40	/	/						
	iter	/	/	/	/						
7	heuristic [pre] (10c)	exp	1	/	Nielsen94b	usability, ease/ pleasantness of use & learning functionality	w port	nov/med	Usability from Two Perspectives—a Study of an Intranet and an Organization [77] - 2005	computer science	MA
	interview [s-f, a rec]	nov/med	19	60	/						
	quest (8q)	nov/med	138	/	/						
	study	nov/med	3	120	/						
8	heuristic (10c)	exp	14	/	Nielsen93, Nielsen94b	usability, ease of learning, satisfaction	mob	nov/exp	Building Usability into Health Informatics—Development and Evaluation of Information Systems for Shared Healthcare [135] - 2007	biomedical informatics	PhD
	quest (13q)	nov	8	/	/						
	study [s rec, v rec] (14t)	nov	8	/	/						
	iter, proto	/	/	/	/						
9	heuristic (22c)	med	4	/	Nielsen94b	design, overall usability	w app	var	Konzeption, Entwicklung und Usability Evaluation einer Webanwendung für die Verwaltung von Webhosting Leistungen (Concept, Development, and Usability Evaluation of a Web Application for the Administration of Webhosting Services) [130] - 2006	computer science in media	MA
	interview [f] (Ø5q)	var	8	/	/						
	study [t-a, a rec] (Ø8t)	var	4	30	/						
	iter	/	/	/	/						
10	quest [prel, survey] (18q)	var	Ø20	/	/	requirements, usability, utility	mob	var	Evaluation, Konzeption und Modellierung eines mobilen Informationssystems mit J2ME für den Einsatz bei Sportveranstaltungen am Beispiel eines Golfturniers (Evaluation, Concept, and Model for a Mobile Information System with J2ME for the Use at Sporting Events using Golfing Tournaments as an Example) [136] - 2006	computer science	MA
	study (5t, 9q)	var	9	/	/						
	walk [cogn]	exp	1	/	/						

CONTINUED ON NEXT PAGE

#	Method	Par	#Par	T/min	Sources	Eval Subject	App	T Users	Thesis	Field	Cat
11	heuristic (9c) study [t-a, f-b] iter	exp med/exp /	1 / /	/ / /	Nielsen93, Pradeep98 / /	usability, design	s-a	med/exp	BALLView, a Molecular Viewer and Modelling Tool [100] - 2007	bio-informatics	PhD
12	guideline (119c) guideline short (52c) study [t-a, a rec, s rec] stats	exp var / /	10 50 / /	/ / / /	EVADIS, Sun, IBM, Apple, ISO9241, W3C Microsoft, LynchHorton99 / /	design, ergonomics	w site	var	Medienergonomische Gestaltung von Online-Informationssystemen des Typs “Register” (Media Ergonomic Design of Online Information Systems, Type “Register”) [129] - 2002	computer science	PhD
13	study [f-b, t-a, v rec, s rec] (3t) walk [cog] iter	med/exp exp /	5 2 /	/ / /	/ / /	design, overall usability	w site	med/exp	A Usability Problem Diagnosis Tool—Development and Formative Evaluation [95] - 2003	computer science	MA
14	guideline study [field, t-a, log] (6t) iter	exp var /	1 16 /	/ / /	Nielsen00, Spool99, IBM, LynchHorton99, Borges01, Rosenfeld98, Fleming98, Thissen01 / /	overall usability	w site	var	Die Verbesserung von WebSites auf der Basis von Web Styleguides, Usability Testing und Logfile Analysen (Enhancing WebSite Usability on the Basis of Web Styleguides, Usability Testing, and Logfile Analysis) [7] - 2001	information science	MA
Only USER-BASED Evaluation Techniques											
15	eye (7t) quest (52q) study [field, f-b] iter, stats	var var var	3 1000 15	/ / 90	/ / /	efficiency, effectiveness, usability, user perception	w port	var	Ergonomie multimedialer interaktiver Lehr- und Lernsysteme (The Ergonomics of Multimedial, Interactive Teaching and Learning Applications) [53] - 2005	computer science	PhD
16	interview [i-f] quest (Ø13q) study [comp, metric, t-a] iter, peer, pilot	med/exp med/exp med/exp /	10 10 10 /	/ / /	/ QUIS /	usability, user perception, system’s strength &weakness	s-a	var	User Interfaces for Accessing Information in Digital Repositories [56] - 2004	computer science	PhD
17	interview quest (18q) study [t-a, a rec, metric] (4t) iter, pilot	med med med /	5 22 22 /	/ / / /	/ MIT, Diaz et al.02 / /	overall usability	w port	var	Usability Evaluation of a Hypermedia System in Higher Education [80] - 2008	computer science	MA
18	ex [metric, t-a] (5t) interview [post, f] (8q) quest [pre, post] (19d, 3q)	med med med	30 30 30	/ / /	/ / /	overall usability, design, task efficiency	s-a	var	Photoware Interface Design for Better Photo Management [91] - 2005	computer science	MA
19	interview [s-f] (38q) quest [remote, field] (Ø15q) study [comp, t-a] (Ø20t) proto	med/exp med/exp med/exp /	37 Ø49 7 /	30-60 / 30-45 /	Yee03 / / /	potential users/ tasks, required features	w app	var	An Internet Search Interface for the Ackland Art Museum Collection Database [9] - 2004	information science	MA
20	interview [s-f] (15q) quest (Ø31q) iter, stats	exp med/exp /	15 Ø103 /	/ / /	/ UIS, QUIS /	user satisfaction with content, usage, and IT support	s-a	var	Einsatz und Evaluierung eines evolutionären IT-Konzepts für ein integriertes klinisches Informationssystem (Application and Evaluation of an Integrated Clinical Information System) [13] - 2007	medical informatics	PhD
21	quest (Ø7q, 4d) study [log] iter, stats	var var /	130 130 /	/ / /	/ / /	design, user preferences, acceptance, overall usability	term	var	Computergestützte Informationssysteme im Museum (Computer-Based Information Systems in the Museum) [103] - 2007	computer science	PhD
22	ex [s rec, v rec, t-a, i-f, metrics, comp, expl, log] (Ø17t) quest (Ø15q) iter, stats	med med /	Ø26 Ø26 /	60-120 / /	/ AttrakDiff, NASA TLX /	performance,	mob	var	Zoomable User Interfaces on Small Screens—Presentation and Interaction Design for Pen-Operated Mobile Devices [17] - 2007	computer science	PhD
23	ex [comp, log, metric] (Ø17t) quest (Ø5d, Ø14q)	med med	Ø16 Ø16	/ /	/ NASA TLX, SUS	usability of magic lens approach (zoom)	s-a	var	AR Magic Lenses: Addressing the Challenge of Focus and Context in Augmented Reality [92] - 2007	computer science	PhD
24	eye quest [field, lab] (Ø32q) study [field, lab, log, t-a] (Ø5t)	/ var var	/ Ø20 Ø18	/ / /	/ IsoNorm, deJong00, IBM, WAMMI, NielsenWeb /	interface quality, dialog behaviour, user support	w site w site	var var	Strategien zu Bewertung der Gebrauchstauglichkeit von interaktiven Web Interfaces (Strategies for Evaluating the Usability of Interactive Web Interfaces) [120] - 2003	computer science	PhD
25	ex [v rec, metric] (1t) quest [pre, post] iter	var med/exp /	54 26 /	60 / /	/ / /	acceptance, efficiency	pen	var	Design und Implementierung einer stiftzentrierten Benutzungsoberfläche (Design and Implementation of a Pen-Based User Interface) [48] - 2001	computer science	PhD

CONTINUED ON NEXT PAGE

#	Method	Par	#Par	T/min	Sources	Eval Subject	App	T Users	Thesis	Field	Cat
26	quest (20q) study [expl] (Ø2t)	nov/med nov/med	30 30	/ 150	WAMMI /	perceived and actual usability	w site	var	An Empirical Foundation for Automated Web Interface Analysis [68] - 2001	computer science	PhD
27	quest (19q) study (6t) pilot	var var /	26 54 /	30 Ø80 /	/ / /	efficiency, ease of use, understandability, GUI	w site	var	A Guide to Improving the E-Commerce User Interface Design [140] - 2005	information science	MA
28	quest (Ø8d, 54q) study [diary, s rec] (18c)	med med	16 16	/ / /	BSMA, IsoNorm, CSUQ, IsoMetrics, AttrakDiff /	ergonomics, effect of adaptivity	s-a	var	Entwicklung einer Methode und Pilotstudie zur Langzeitevaluation von adaptiven User Interface Elementen (Developing an Approach for Long-Term Evaluation of Adaptive User Interface Elements, and Pilot Study) [57] - 2004	information science	MA
29	study [metric, comp] (Ø6t)	var	Ø12	30-45	/	learnability, flexibility	s-a	var	A User Interface for Coordinating Visualizations Based on Relational Schemata: Snap-Together Visualization [119] - 2000	computer science	PhD
30	study (5t) peer, proto	med	8	120	/	ease of use, pleasantness	mob	var	User Interface Design and Usability Testing of a Podcast Interface [74] - 2007	communication science	MA
31	interview [f] (54q, 4d) pilot	nov/med	11	30	/	usability, accessability, learning success	w port	nov/med	Evaluation des Lernerfolgs einer Blended Learning Maßnahme unter Berücksichtigung der Barrierefreiheit (Evaluating the Learning Success of a Blended Learning Method, Considering Accessibility) [37] - 2007	computer science in media	MA
32	quest (50q)	exp	1	/	HDEQ	overall usability	w app	var	Konzipierung und Implementierung einer Online Hilfe für ein virtuelles Konferenzsystem im Rahmen des von der Europäischen Gemeinschaft geförderten Projektes “Invite EU” (Conception and Implementation of an Online Help System for a Virtual Conference System within the Project “Invite EU”, Funded by the European Commission) [96] - 2000	computer science	MA
Only EXPERT-BASED Evaluation Techniques											
33	walk [plur] (41q)	var	15	/	Nielsen93, ISO9241, Constantine	overall usability	w app	med/exp	Usability von Web Content Management Systemen - Analyse von Verbesserungspotentialen im Bereich der Usability (Usability of Content Management Systems—Analyzing Potential Usability Enhancements) [155] - 2006	computer science	MA
34	heuristic (13c)	var	8	/	Nielsen94b, Pierotti95	overall usability	w port	var	A Course Content Management System Development and its Usability [79] - 2004	computer science	MA
35	guideline [comp] (97c)	exp	1	/	Nielsen93, Nielsen00, Krug00, NielsenTahir02, Pearrow00, Baker01, Thissen01, Manhartsberger01	overall usability	w site	var	Eine vergleichende Analyse der Websites von Anbietern pneumatischer Automatisierungskomponenten - heuristische Usability Evaluation und zielbasierte Content Analyse (A Comparative Analysis of Websites of Pneumatic Automation Component Suppliers— Heuristic Usability Analysis and Goal-Based Content-Analysis) [31] - 2002	information management	MA

Table 6: Theses and their Field of Publication

Field	Theses
Ph.D. Theses	
computer science	#1, #3, #12, #15, #16, #21, #22, #23, #24, #25, #26, #29
bio-informatics	#11
biomedical informatics	#8
medical informatics	#20
MA Theses	
computer science	#2, #6, #7, #10, #13, #17, #18, #32, #33, #34
communication science	#30
computer science in media	#9, #31
information management	#35
information science	#5, #14, #19, #27, #28
media & communication science	#4

3.2 Analysis of the findings

In this section, the results of the analysis of the collected theses are discussed. Subsection 3.2.1 briefly introduces the investigated theses (35 in total—15 Ph.D. and 20 MA). The results of the examination of the used evaluation techniques are provided in subsection 3.2.2. Subsection 3.2.3 finally summarizes the sources, that were used as a basis for the evaluations in the examined theses.

3.2.1 Examined Papers

Within this section, the examined theses are introduced shortly with the goal, to provide the most important background facts. In total, 15 Ph.D. theses were examined, containing a total of 12 theses directly from computer science, and 3 theses from interdisciplinary fields. We further investigated a total of 20 MA theses, 10 theses directly from computer science, and 10 theses from interdisciplinary or strongly related fields. The different specific fields and the corresponding theses are listed in Table 6.

In the following we provide a summary of some general information about each thesis: consecutive number, as also used in Table 5, title, year, as well as a short description of the evaluated system(s). The latter might be of interest for researchers planning an evaluation themselves. The theses are listed in the same ordering as they appear in the synoptical table (Table 5), that is, theses that used both user-based and expert evaluations first, those that applied only user-based techniques second, and finally those that used only expert evaluation techniques.

Both user-based and expert evaluation

- #1 *A Measure of Fun—Extending the Scope of Web Usability* [159] - 2003. Existing web usability measures were extended to evluate specifically entertainment-centered website in terms of joy of use and entertainment factor.
- #2 *Redesigning User Interfaces by the Means of Navigation* [78] - 2003. A new navigational approach was developed for IBM's DB2 Performance Expert Client, and evaluated, focussing on acceptance of the new approach and user satisfaction.
- #3 *Designing Ergonomics into Web Presences: Analyzing Human Behaviour while Using the Web, and User-Specific Design Suggestions* [45] - 2004. The design of websites was evaluated in terms of ergonomics specifically for seniors.
- #4 *Use and Usability of the Mountain Information System of the DAV—a Usability Study* [59] - 2007. The rock information system of the DAV (German Alpin Association), was evaluated with special focus on its understandability, navigation structure, overall usability, and the quality of its contents.
- #5 *Usability in the Context of Managers, Developers and Users of the Website of the Central Library at the University of Brazil* [30] - 2006. The website of the Central Library at the University of Brasilia was assessed in terms of its overall usability.

- #6 *Evaluation of the User Interface of a Web Application Platform* [157] - 2006. The web-platform Content Studio, a platform for collaboration and content management, was evaluated with a special focus on the design of its GUI, and potential malfuntions; though, overall usability was considered, too.
- #7 *Usability from Two Perspectives—a Study of an Intranet and an Organisation* [77] - 2005. The intranet site of the company OMX was evaluated with a special focus of ease and pleasentness to use, ease of learning, and its functionalities.
- #8 *Buildung Usability Into Health Informatics—Development and Evaluation of Information Systems for Shared Homecare* [135] - 2007. A mobile health recording system for usage with PDAs and tablet PCs was implemented, and evaluated in terms of its overall usability, effectiveness, and ease of learning.
- #9 *Conception, Development, and Usability Evaluation of a Web Application for the Administration of Webhosting Services* [130] - 2006. A web application for administrating webhosting services was developed and evaluated, especially focussing on its design, but also considering overall usability.
- #10 *Evaluation, Concept, and Model for a Mobile Information System with J2ME for the Use at Sporting Events Using Golfing Tournaments as an Example* [136] - 2006. A mobile information system for use with e.g. PDA's as information tool at sporting events was developed, and evaluated in terms of its overall usability and utility.
- #11 *BALLView, a Molecular Viewer and Modelling Tool* [100] - 2007. An intuitively usable application, that combines the functionalities for creating and visualizing molecular models was developed, with an emphasis on the overall usability and the GUI design of the tool.
- #12 *Media Ergonomic Design of Online Information Systems, Type "Register"* [129] - 2002 . The website of the city of Bremen, Germany, was examined, newly designed, and evaluated in terms of design and ergonomics.
- #13 *A Usability Problem Diagnosis Tool—Development and Formative Evaluation* [95] - 2003. A web- and knowledge-based tool for automated support of usability problem diagnosis was developed and evaluated in terms of its own usability.
- #14 *Enhancing Website Usability on the Basis of Web Styleguides, Usability Testing and Logfile Analysis* [7] - 2001. The website of a summer cottage agency was newly designed on the basis of guidelines, and its overall usability was assessed.

——————— *Only user-based evaluation* ———————

- #15 *The Ergonomics of Multimedial, Interactive Teaching and Learning applications* [53] - 2005. The web portal of a virtual college of higher education was evaluated in terms of its overall usability, efficiency, effectiveness, and the users' perception.
- #16 *User Interfaces for Accessing Information in Digital Repositories* [56] - 2004. Two database frontends for exploring large digital information repositories were implemented, and evaluated in terms the overall usability and the users' perception of the interface.
- #17 *Usability Evaluation of a Hypermedia System in Higher Education* [80] - 2008. An educational wiki system for higher education was evaluated and newly designed in terms of its overall usability.
- #18 *Photoware Interface Design for better Photo Management* [91] - 2005. Two features for existing photoware interfaces were developed, and evaluated in terms of usability, especially focusing on interface design and efficiency.

- #19 *An Internet Search Interface for the Ackland Art Museum Collection Database* [9] - 2004. A web-based database frontend for the Ackland art museum was developed and evaluated in terms of its usability.
- #20 *Application and Evaluation of an Evolutionary IT Concept for an Integrated Clinical Information System* [13] - 2007. A distributed clinical software system, providing features for e.g. organizing patients' records, was developed, and the users' satisfaction with content and functionalities, the usage, and the IT support of the system were investigated.
- #21 *Computer-Based Information Systems in the Museum* [103] - 2007. An information terminal system for the Senckenberg Museum Frankfurt, Germany, was developed, and evaluated, mainly regarding the basic acceptance of the system, and its overall design and usability.
- #22 *Zoomable User Interfaces on Small Screens—Presentation and Interaction Design for Pen-Operated Mobile Devices* [17] - 2007. The applicability of starfield displays and zoomable map-based interfaces for mobile devices was investigated, the overall usability, users' workload and satisfaction were the main focus of the evaluation.
- #23 *AR Magic Lensess: Addressing the Challenge of Focus and Context in Augemented Reality* [92] - 2007. A two-dimensional zooming approach, "magic lenses", was implemented for an GIS (geographic information system), and assessed in terms of its overall usability.
- #24 *Strategies for Evaluating the Usability of Interactive Web Interfaces* [120] - 2003. An approach for website usability evaluation was developed and probed for different websites.
- #25 *Design and Implementation of a Pen-based User Interface* [48] - 2001. A pen-based user interface approach was developed, and several forms of select-actions, as well as acceptance and efficiency of such an interface was evaluated.
- #26 *An Empirical Foundation for Automated Web Interface Analysis* [68] - 2001. A new approach for automated website analysis was developed, which included evaluating the usability of selected websites and redesigning them following the newly developed approach.
- #27 *A Guide to Improving the E-Commerce User Interface Design* [140] - 2005. Four distinct e-commerce websites were evaluated in terms of their ease of use, efficiency, ease of understanding, and design of the user interface.
- #28 *Developing an Approach for Long-Term Evaluation of Adaptive User Interface Elements, and Pilot Study* [57] - 2004. Existing office software was evaluated in terms of the usability of adaptive user interface elements.
- #29 *A User Interface for Coordinating Visualizations Based on Relational Schemata: Snap-Together Visualization* [119] - 2000. A user interface for creating custom data exploration interfaces and visualization was developed and its ease of learning, ease of use, and flexibility were evaluated.
- #30 *User Interface Design and Usability Testing of a Podcast Interface* [74] - 2007. Podcasting software, intended to be used with mobile phones or PDAs, was developed, and its usability was evaluated, especially focussing on ease of use and pleasentness.
- #31 *Evaluating the Learning Success of a Blended Learning Method, Considering Accessibility* [37] - 2007. A blended learning approach, intended especially for disabled people, was evaluated in terms of its usability, accessability, and the potential learning success of target users.

- #32 *Conception and Implementation of an Online Help System for a Virtual Conference System within the Project "Invite EU", Funded by the European Commission* [96] - 2000. An online help system was developed, and evaluated on the basis of the Help Design Evaluation Questionnaire (HDEQ [38]) with focus on its overall usability.

Only expert evaluation

- #33 *Usability of Content Management Systems—Analyzing Potential Usability Enhancements* [155] - 2006. Possible guidelines for an evaluation of web content management systems were prepared, and four existing and actually used content management systems were evaluated on basis of these guidelines.
- #34 *A Course Content Management System Development and its Usability* [79] - 2004. A content management system for organizing and providing course information for students was developed and its overall usability and effectiveness assessed.
- #35 *A Comparative Analysis of Websites of Pneumatic Automation Component Suppliers* [31] - 2002. Websites of pneumatic component suppliers were comparatively evaluated through the means of heuristic evaluation and goal-based content analysis.

3.2.2 Analysis of Applied Evaluation Techniques

This section provides the results of investigating the evaluation techniques that were reported in the examined works. Basically we inculded only techniques, specifically designed for usability evaluation—as described in sections 2.1 to 2.3—for the analysis. Indeed we also found that general techniques—for example, prototyping, pilot testing, peer reviewing (see section 2.4)—were actually applied, too. Therefore we listed them in the synoptical table (Table 5) for reasons of completeness; yet they will not be further discussed to preserve the focus on usability evaluation and its applicability.

Concerning the actually applied evaluations, we analyzed the following aspects: the application types that were evaluated, the actual evaluation techniques, additional features used with the basic evaluation techniques, the number of participants, the expertise of target users and evaluators, the number of items—for example, questions, tasks, heuristics—used for the evaluation, the mean duration of the evaluation, and finally the sources—for example, technical literature—that were used to develop the evaluation.

Application Types

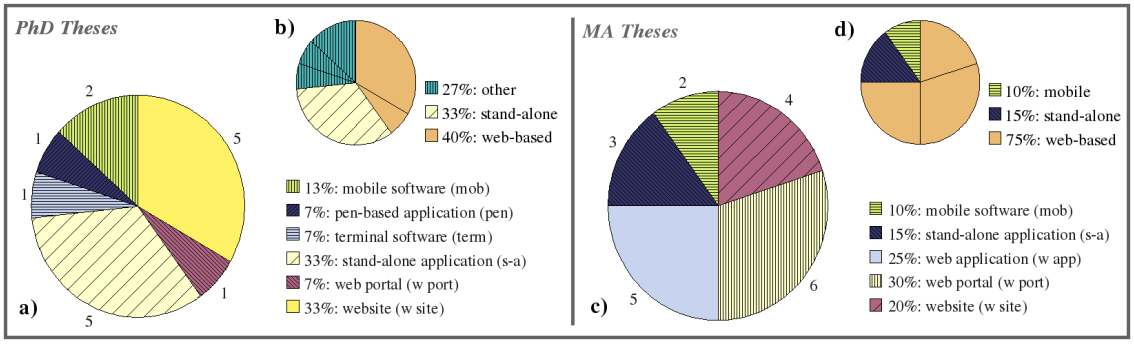

Figure 2: Application Types, as Evaluated wtihin Ph.D. theses (a/b) and MA Theses (c/d)

First we investigated the different application types, that were evaluated within the examined theses. Figure 2 depicts the distribution of the different application types found. The lefthand

side of the figure—a) and b)—shows the findings for the set of Ph.D. theses, whereas the findings for the MA theses are presented in the righthand part of the figure—c) and d). Basically, the detailed distribution of all application types found is presented in the bigger pie charts, a) and c). In contrast to that, the smaller charts—b) and d)—provide a summarized view. Therefore, all similar types were merged into a more general category—for example, all application types someway related to the web were merged into category web-based.

What is noticeable for both Ph.D. and MA theses is the fact, that web-based applications were investigated and evaluated most frequently: within 40% of the Ph.D. theses and 75% of the MA theses. Whereas in MA theses the distribution between distinct web-based application types was nearly equal—websites 4 times, web applications 5 times, web portals 6 times—, in Ph.D. theses the type web portal was investigated just once, but websites in 5 cases. Also in both Ph.D. and MA theses, stand-alone application was the application type second most frequently evaluated: 5 times in Ph.D and 3 times in MA theses. Mobile applications were investigated in both 2 Ph.D. and MA theses, whereas pen-based interfaces and terminal software was only considered each once in a Ph.D. thesis. The type web application in turn was only considered within MA theses.

Evaluation Techniques

This section describes, which evalution techniques were acutally applied within the investigated theses. Figure 3 presents an overview of the techniques and their frequency of usage. The distinct techniques are listed in alphabetical order along the Y-Axis, their frequency of usage along the X-Axis. Some researchers applied an evaluation technique more than once sometimes, so it has to be remarked, that we expressed the frequency of usage through the number of distinct theses, a technique was applied in. This corresponds to the presentation in the synoptical table (Table 5) where we listed each technique only once per thesis, too. For each technique at most two bars are provided, representing the number of Ph.D. and/or MA theses. Light-colored bars represent Ph.D. theses, dark-colored bars MA theses. The X-Axis is labeled in steps of two, so the exact number is additionally presented next to the corresponding bar.

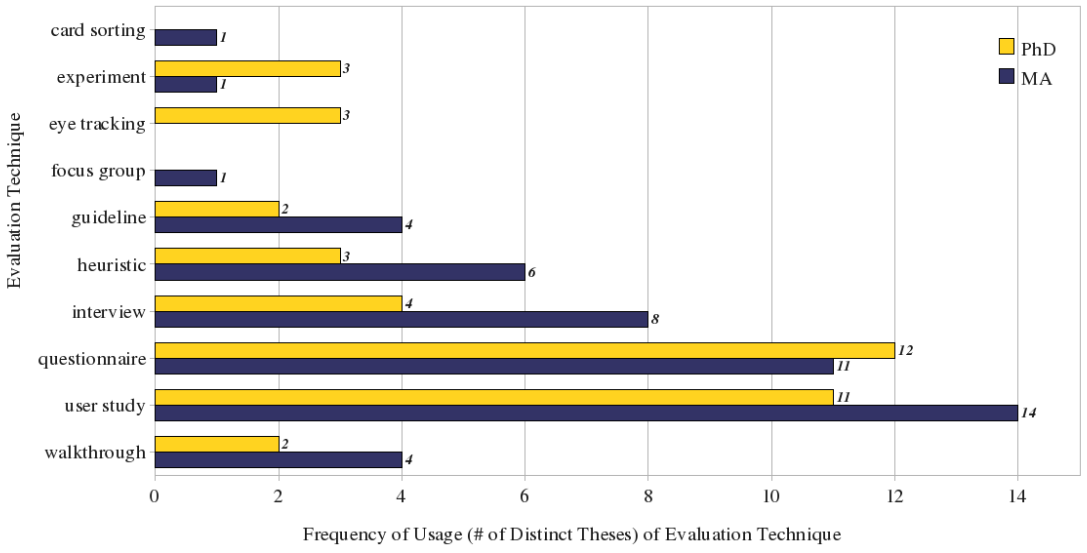

Figure 3: Evaluation Techniques

Both for Ph.D. and for MA theses, user study, questionnaire and interview were the most frequently used techniques. This may be due to the fact, that those approaches already belong to the more established evaluation techniques in computer science and related fields today - the techniques, one would think of first, when an evaluation has to be conducted. Moreover, theses from computer science often do not solely focus on evaluation issues. Most frequently also one or

more practical tasks (like, for example, developing a piece of software, (re-)designing interfaces and the like) are parts of the work, so evaluation mostly is just one of several duties of the researcher. This might give reasons for chosing evaluation techniques that are on the one hand well established, and on the other hand require manageable efforts.

Furthermore, interview, heuristic evaluation, guideline-based evaluation, and walkthrough were—compared to Ph.D. theses—somewhat more frequently applied within MA theses. Also, two additional techniques—focus group evaluation and card sorting—that were not found within the Ph.D. dissertations, were used once in a MA thesis each.

The controlled experiment—though also quite well-researched, and in fact often applied in other fields, for example, psychology—was quite rarely applied in the context we investigated: within 3 Ph.D. theses and just 1 MA thesis. A possible explanation might be the complex procedures required, that result in increased financial and organizational efforts for planning and conducting the evaluation. Eye-tracking also was applied just in the case of 3 Ph.D theses. This might be due to the fact, that this technique itself is rather young, and its applicability and concrete procedures for usability evaluation are still being researched to date. Moreover, the hardware required for conducting an eye tracking session is still not broadly distributed.

Alltogether, a total of 31 user-based evaluations and only 7 expert evaluations were conducted within the set of Ph.D. dissertations, whereas 36 user-based evaluations and 14 expert evaluations were applied within MA theses. The fact, that clearly more user-based evaluations were applied might reflect the common trend to incorporate users and their preferences and needs more often into the design and evaluation of a system than in the past. One remarkable finding concerns the combined usage of evaluation techniques. As Nielsen [106] already suggested earlier, combining user-based and expert evaluations can yield most valuable results as not only they can detect distinct types of usability problems, but user testing can also be conducted more effectively when the interface has been designed or revised on the basis of an expert evaluation in advance. We found that in several cases the researchers complied with this suggestion, as a combination of user-based and expert evaluations was applied within 5 Ph.D. theses and 9 MA theses.

We also found another recommendation—to augment user studies or similar techniques with some kind of questioning to gain additional, detailed, and subjective user information—adhered to. Thus the combination of user study or the controlled experiment with further questioning through questionnaire or interview was applied within 11 Ph.D. theses and 11 MA theses.

Another general finding was, that in most theses—both in Ph.D. and MA theses—more than one distinct techniques was applied for the evaluation. Only in 1 Ph.D. and 6 MA theses a single technique was used.

Additional Features for Evaluation Techniques

Apart from the basic techniques we also looked at the additional features that were used in combination with the evaluation methods. This includes features such as video recording or thinking aloud; on the other hand, techniques that were categorized as hypbrid approaches in section 2 were also treated as additional features. The latter was due to the fact, that those actually applied hybrid techniques—for example, remote testing and comparative analysis—could be used in combination with several distinct basic techniques. Thus we considered it easier to treat those techniques just like the additional features. Figure 4 presents the frequency of usage of the features, which was expressed in the number of distinct evaluations, a feature was applied. The features are displayed along the Y-axis of the chart, whereas the frequency is expressed along the X-axis. Again we provided at most two bars per feature, representing the frequency as found within the set of Ph.D. theses (light-colored bars) and within the set of MA theses (dark-colored bars). The exact number of times a feature was used is expressed through the number next to the bars. Basically it has to be noted that in many cases no information about the usage of additional features was provided by the authors of the investigated theses. Thus, the numbers presented in Figure 4 can only illustrate a general tendency.

The two hybrid approaches comparative analysis and remote evaluation—represented by the uppermost bars in the chart—were actually applied in only some cases. The chart further depicts

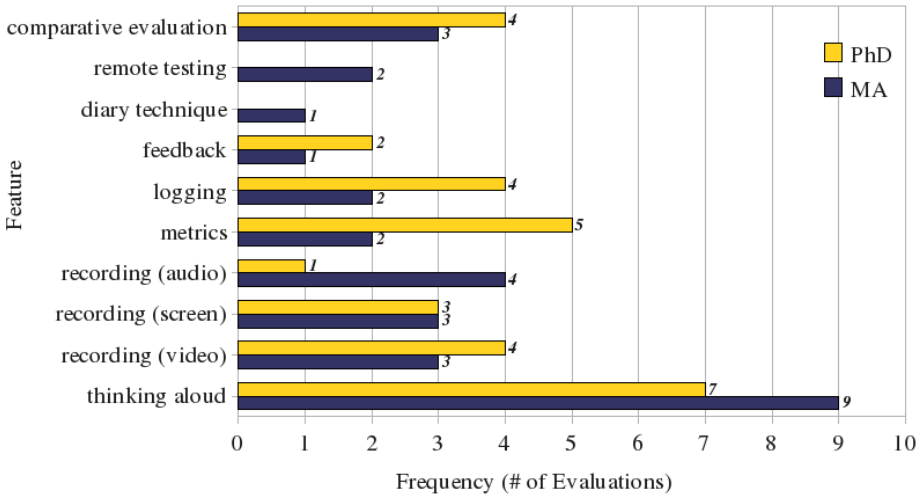

Figure 4: Usage of Supplementary Features

that the technique of thinking aloud was applied quite often, both within Ph.D. and MA theses. Also quite often used was recording as a protocolling technique: in a total of 8 Ph.D. theses and 10 MA theses one of the three recording variants—audio recording, screen recording, and video recording—was applied. Finally the logging of user actions as well as the measurement of metrics were by trend applied somewhat more frequently within Ph.D theses than within MA theses.

Number of Participants

The next aspect we considered interesting is the average number of participants recruited for the evaluations, where participants included both expert evaluators and test users, depending on the evaluation technique. To gain some general insight about the numbers participants in real-world evaluations within our investigated context we calculated an average value per thesis. That is, we summed up the number of participants of all distinct evaluation techniques applied within a thesis and then divided by the number of techniques used per thesis. This resulted in an average number of participants for each thesis we investigated.

Figure 5 depicts the calculated average numbers for each thesis with each bar representing the number of participants for exactly one thesis. The X-axis is labeled by the consecutive numbers of the 35 examined theses. The representative bars are plotted in ascending order by the calculated number of participants along the X-axis. The Y-axis is labeled by the average number of participants. The exact number of participants is additionally provided above the bar representing the number for each thesis. As before, the bars representing Ph.D. theses are presented in a light color, the bars representing MA theses in a darker color.

A general tendency that can be observed is the rather small number of participants in many cases. In a total of 14 theses—4 Ph.D. theses and 10 MA theses—we found less than or exactly 10 participants recruited, which conforms to about 40% of the examined theses. The largest proportion was made up by numbers of participants between more than 10 but less than 50. In total, 18 theses—8 Ph.D. theses and 10 MA theses—, that is about 51% of all theses, fell into this category. In the 3 remaining Ph.D. theses a higher number of participants was recruited. In each of these 3 cases, the reason for the higher numbers was the application of questionnaires for which a higher number of participants was recruited. Finally we calculated the average number of participants for all Ph.D. and MA theses. Therefore, we summed up the average numbers for all Ph.D./MA theses and divided the value by the total number of Ph.D./MA theses. This resulted in a total average number of participants of about 50 ($= \frac{748}{15}$) participants for the Ph.D and about 16 ($= \frac{319}{20}$) participants for the MA theses. A possible reason for the rather small number of participants might have been the costly (financial/organizational) effort an extensive evaluation would require. Therefore

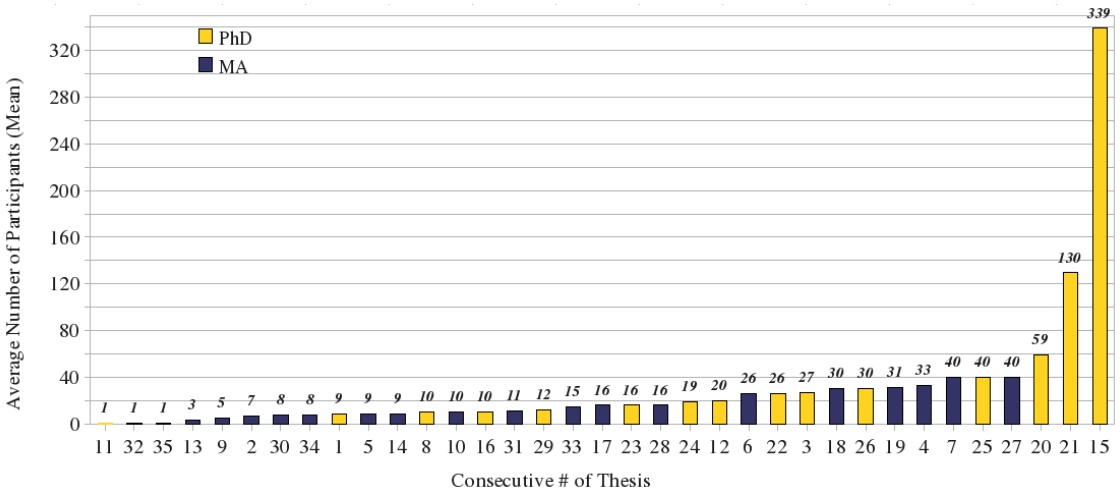

Figure 5: Number of Participants per Thesis

a reasonable way seemed to be to evaluate with smaller samples of users, mostly recruited from the direct environnement of the resarcher (school/university, research facility, or organization). Moreover, all those "smaller" evaluations were complemented by at least one distinct, additional technique, which could compensate for the rather small number of participants in some respects.

Apart from the average number of participants per thesis we were further interested in the average number of participants per evaluation technique, depicted in Figure 6. The different techniques are listed along the Y-axis in alphabetical order, whereas the X-axis is marked by the number of participants. Here again, at most two bars are presented for each technique, one (light-colored) representing the number of participants within Ph.D. theses and one (dark-colored) representing the number of participants within MA theses. The exact number is additionally provided next to each bar. The bar chart in Figure 6 illustrates, that for the questionnaire technique the most participants

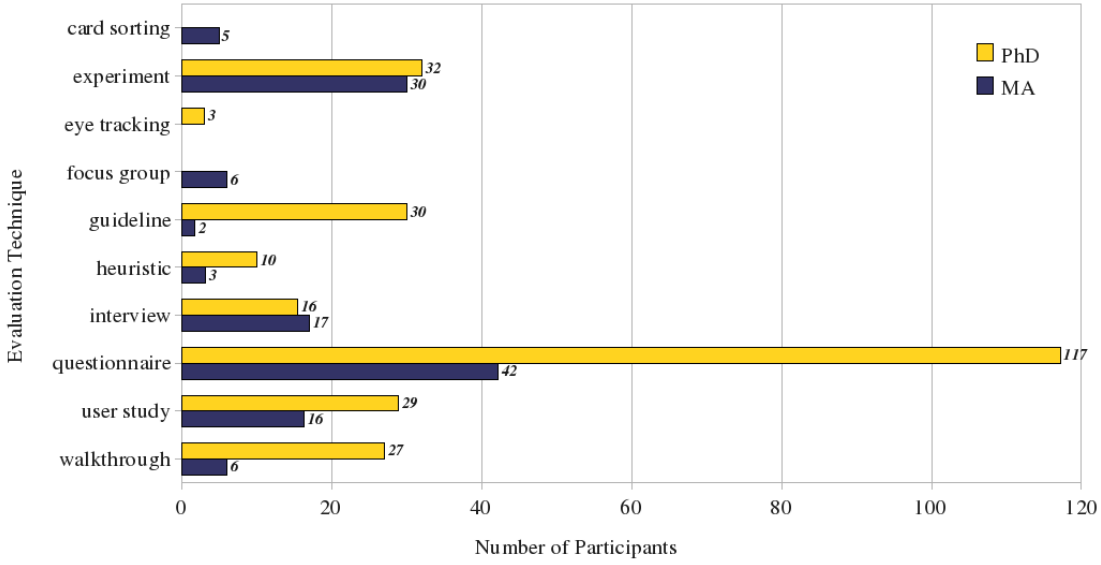

Figure 6: Number of Participants per Evaluation Technique

were recruited, both in Ph.D. and MA theses. Also for the experiment, the interview technique,

and the user study a still rather high number of participants was found. A bit surprising is the result for the heuristic evaluation—as this is a expert evaluation technique, one would rather expect a small number (say, about 1 to 3) of evaluators. Yet, in the case of the Ph.D. theses, the average number of participants was 10. This again conforms to Nielsen's [106, p.156] recommendation to conduct a heuristic evaluation with at least 3 to 5 evaluators, as several evaluators are likely to find more and distinct problems.

Expertise of Target Users and Evaluators

Another interesting aspect of an evaluation is the expertise—that is, the knowledge and/or the prior experience with the system—of the target users and the evaluators (expert evaluators or test users) of the assessed system.

The target users of the systems evalauted both in Ph.D. and MA theses were in the majority of cases of the type *various (var)*. Only two Ph.D. theses constrained the target user group—thesis # 8 to novice/experienced, and thesis # 11 to intermediate/experienced. Likewise, 4 MA theses constrained the target users—theses # 7 and # 31 to nov/med, and theses # 13 and # 33 to med/exp. As even theses constrained target user groups vary within a smaller range of experience, the systems evaluated were all intended to be used by broader user groups, not specified in detail.

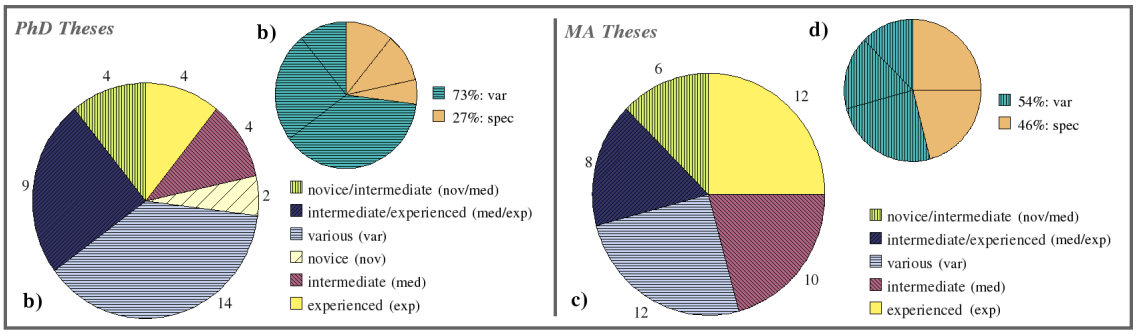

Figure 7: Expertise of Evaluators

The distribution of the different levels of expertise of the evaluators is depicted in Figure 7. Here, the results for the set of Ph.D. theses is presented on the lefthand side of the figure—a) and b)—and the results for the set of MA theses on the righthand side—c) and d). Thereby, the smaller pie charts, b) and d) present a summarized view of the pie charts a) and c), merging all mixed expertise types—such as med/exp—into one summarizing category *var (various)*, and all specific types—such as nov—into the category *spec (specific)*. The charts show, that there exists a strong tendency, to have more than one type of participant. 73% of the Ph.D. theses and 54% of the MA theses recruited participants with a mixed expertise. Moreover we found, that in quite a small number of theses novices were included within the group of participants.

Number of Items

The next value we examined, was the average number of items—that is, (demographic) questions, tasks, heuristics, or guidelines—used throughout the evaluations. We considered it most informative, to calculate separate values for each type. As not all authors explicitly described accurate values here, we used the number of evaluations, where values for the specific type were given, as the basis for calculating the average value for this type, that is, excluding those works, not providing an appropriate value. For example in a total of 4 MA theses a guideline-based evaluation was conducted, but only three researchers provided the number of guidelines used; thus 3 is the basis in this case. Summing up the values of the number of guidelines, we receive a value of 319 (=113+109+97). Dividing this by the basis of 3 results in an rounded average number of 106 guidelines used within the set of MA theses.

Table 7: Average Number of Items Used

	Ph.D.	MA
demographic questions	7	10
guidelines	86	106
heuristics	10	13
questions	21	26
tasks	8	8
TOTAL average	26	33

Table 8: Average Duration of Evaluation Sessions in Minutes

	Ph.D.	MA
minimum duration	64	71
maximum duration	78	88
TOTAL average	71	80

As Table 7 depicts, the results for each the set of Ph.D. theses and MA theses were quite similar. The average number of tasks was in both cases 8, but also the values for (demographic) questions and heuristics were close together. The only apparent difference was found in the number of guidelines, used for a guideline-based evaluation. Here, the average number for MA theses was about one fourth greater than the number for Ph.D. theses.

It has to be noted, that in two cases—MA theses #4 and #35—the authors themselves originally described their evaluation as a heuristic evaluation. In contrast to that, we classified the techniques as guideline-based evaluation as the authors assembled a rather extensive and set of detailed guidelines which better fits our description and classification of guideline-based evaluation.

Mean Duration

In many cases no information on the duration of the evaluation sessions was provided by the researchers. Thus again we included only those theses in the calculation, that provided appropriate values. As often time spans were described rather than absolute time values, we calculated both an average minimum and an average maximum value for the duration of evaluations.

Table 8 shows that the average duration of evalaution sessions in Ph.D. theses is quite close to the values found for MA theses, but with the tendency of being slightly shorter. The calculated total average duration for Ph.D. theses was 71 ($= \frac{64+78}{2}$) minutes and 80 ($= \frac{71+88}{2}$) minutes for MA theses . Alltogether, evaluations thus lasted 76 ($= \frac{71+80}{2}$) minutes, that is, between one and one and a half hours.

Sources

Most researchers of the investigated theses used one or several sources as a basis to develop their own evaluation approaches. These sources included technical literature, basic procedures, or predefined guidelines and checklists, heuristics, or questionnaires.

Both for Ph.D. theses and MA theses we found, that many distinct sources—mostly originating from comuter science or strongly related fields—were used. An entire listing of all basic works is provided in section 3.2.3. Standing out, though, was the fact that the 10 heuristics of Jakob Nielsen [106, 108] were utilized either in their original form, or slightly modified, in 3 Ph.D. theses and 6 MA theses for conducting a heuristic evaluation. This confirms the assumption, that Nielsen's 10 heuristics are not only widely known, but also flexibly applicable in various contexts or at least a good basis to develop own specific heuristics. This is supported by the fact, that in case of the Ph.D. theses, the heuristics were applied for the evaluation of three distinct application types: stand-alone applications, mobile software, and websites; in the case of the MA theses, they were applied for assessing websites, web portals, and web applications. Also used both in Ph.D. as well as in MA evaluations were the IsoNorm questionnaire [128, 127], the ISO 9241 norms [34] or technical literature from Lynch & Horton [94].

Within the set of Ph.D. theses, we found each of the QUIS, the WAMMI, and the NASA TLX used as a basis for the development of user questioning in two works, the IBM interface and webdesign guidelines as a basis for questioning and guideline development each once.

Concerning the set of MA theses, it is remarkable that apart from Nielsen's 10 heuristics, we also found that several other sources of Nielsen had been used. Thus, a set of 113 guidelines for website design, proposed by Nielsen & Tahir [116] in their book "Homepage Usability", also served as a foundation for developing tailored heuristics. Moreover, this book was also used as the basis for the development of two guideline-based evaluations.

The remaining sources—each of which is listed in Table 9—were only found to be used once.

3.2.3 Literature on Evaluation Techniques

As already mentioned, the authors of the examinded works based their evaluations on several sources: basic procedures and established suggestions—for example, questionnaires or guidelines—or technical literature on design and usability. These were listed in column *Source* of the synoptical table. This section gives an overview of all the applied sources. Describing each one textually would go beyond the scope of this survey so the most important facts were summarized in Table 9.

Sources were grouped by the evaluation techniques they were used to implement (Diaz' listing of evaluation criteria [33], for example, is assigned to the questionnaire-section of the table as it was used to develop an evaluation questionnaire—even though it is originally not a predefined kind of questionnaire). Some sources appear in more than one section of the table, indicating that differnt authors used the same basics for implementing distinct evaluation techniques.

The *Name* column of Table 9 provides the name of each source, as used in the synoptical table. Moreover it contains a reference for further information. The second column lists a short summary of the main subject each source concerns, and, where available, the number of items—for example, questions, heuristics, or the like.

4 Discussion

In this paper, we presented a survey of evaluation techniques for assessing the design and usability of an interface or system. An overview of all evaluation techniques we reported on was provided by Figure 1 in section 2. It has to be noted, that there exist several approaches that aim at integrating several individual evaluation techniques into one comprehensive approach, often also providing specific calculation procedures or tools. Examples are the EVADIS II [121] evaluation compendium and the MUSiC [10] methodology. As those are based on the individual techniques surveyed in this paper, we did not further inspect such approaches but focussed on the basic techniques.

We further discussed the actual application of the surveyed techniques within a specific context: applied scientific works that are characterized by a limited budget of time and/or money for an evaluation in the specific fields of computer science and strongly related. This characteristics were fulfilled best by Ph.D. and MA theses, which is the reason, why we focussed our analysis on those works. Section 3 presented the results of the investigation of the theses. We first summarized the examined works (Table 5 in section 3.1), a total of 35 theses—15 Ph.D. theses and 20 MA theses. The reported usage of evaluation methods was analyzed and described in section 3.2. Additionally, we provided an overview of all sources that were used by the authors of the theses as a basis for developing their evaluation approaches in section 3.2.3 in Table 9.

There were several interesting findings we learned from our analysis:

- Many different types of systems and their interfaces were evaluated in terms of their usability. This suggests, that the basic usability evaluation techniques are applicable in a flexible manner.
- The target user population of the evaluated systems consisted mostly of users with differing knowledge about and/or experience with the system.

Table 9: Literature on Evaluation Techniques

Name	Description
Questionnaire	
AttrakDiff [54, 55]	questionnaire, user satisfaction—28 items
BSMA [41]	questionnaire, mental workload—1 scale
CSUQ [89]	questionnaire, computer system usability—19 items
DATech [28]	handbook, framework for usability engineering evaluation—239 pages
deJong00 [29]	how to characterize heuristics for web communication
Diaz et al.2002 [33]	evaluation criteria for hypermedia systems—12 criteria
HDEQ [38]	questionnaire, evaluation of online help systems—50 items
IsoMetrics [47, 49, 60]	questionnaire, ISO 9241-10 -based evaluation—current vers. 75 items
IsoNorm [90, 127, 128]	questionnaire, software evaluation in terms of ISO 9241-10—35 items
LPS [62]	intelligence test covering 7 primary mental abilities—15 complex subtests
MIT [65]	usability guidelines—62 items
NASA TLX [50, 51, 104]	questionnaire, workload—6 items
NielsenWeb [105]	Nielsen's website, information on (web)usability
QUIS v.7.0 [21, 118, 139]	questionnaire, subjective user satisfaction—current vers. 41 items (short)/111 items
SUS [15, 73]	questionnaire, overall system usability—10 items
UIS [6, 67]	questionnaire, subjective measure of system success—26 items (short)
WAMMI [22, 82]	questionnaire, user satisfaction of websites—current vers. 20 + 28 (optional) items
Interview	
Constantine [24, 25, 26]	book/articles covering web & interface design
ISO9241 [34]	norms on ergonomic interface design—17 norms
Yee03 [161]	study on a basic usability interview methodology
Heuristics	
Baker01 [5]	guidelines for designing websites with flash
Dias01 [32]	usability evaluation of corporate portals
Krug00 [84]	book, website design
Manhartsberger01 [97]	book, website usability
Nielsen93 [106]	book, usability engineering and evaluation—10 heuristics, original
Nielsen94b [108]	book, usability inspection methods—10 heuristics, revised
Nielsen00 [111]	book, website usability
NielsenTahir02 [116]	book, website usability—113 guidelines
Norman88 [117]	book, general issues about designing things
Pearrow00 [124]	book, website usability
Pierotti95 [125]	Xerox' checklist for heuristic evaluation—295 items
Pradeep98 [126]	book, user centered information design for usability
Shneiderman04 [139]	book, interface design—8 heuristics
Thissen01 [150]	book, interface design
Tognazzini [151]	principles of interaction design—16 heuristics
Wroblewski01 [160]	design guidelines for web-based applications—19 guidelines
Guidelines and Checklists	
Apple [1, 2, 3, 4]	interface and web design guidelines
Bastien & Scapin1993 [8]	ergonomic criteria for interface evaluation
Borges98 [14, 43]	web design guidelines
Constantine [24, 25, 26]	web & interface design
ErgoList [154]	ergonomics guidelines and checklists
EVADIS [121]	guide for evaluating software ergonomics
Fleming98 [42]	web site navigation
IBM [63] [64]	interface and web design guidelines
ISO9241 [34]	norms on ergonomic interface design—17 norms
Lynch&Horton99 [94]	book, website design
Microsoft95 [98]	software interface guidelines
Nielsen00 [111]	book, website usability
Parizotto97 [122]	technology & science info. webservices styleguide
Rosenfeld98 [131]	book, webdesign
Spool99 [142]	book, website usability
Sun [145, 146, 147]	interface and web design guidelines
Thissen01 [150]	book, interface design
W3C [156]	web design and accessibility guidelines

- Most evaluations were combinations of at least two distinct techniques. This conforms to suggestions proposed by well-known usability experts (for example Nielsen [106] and Koutsabasis [83]), that a comprehensive usability evaluation requires the application of more than one evaluation technique. Single techniques were applied in a total of only 7 out of 35 theses.
- The techniques most commonly applied in the field of computer science were user study, query techniques, and heuristic evaluation. Thus they are assumedly applicable in a flexible and easy way, and require quite manageable financial and/or organizational efforts.
- In the majority of cases, a rather small number of participants was recruited for most evaluations. The average number of participants within Ph.D. theses was 50, and 16 within MA theses.
- In many cases, the group of participants consisted often of persons possessing a mixed expertise, that is, differing knowledge about and/or experience with the system.
- A rather small numbers of items (questions, tasks, heuristics) was applied throughout the evaluations (see also Table 7). An exception is the number of guidelines, as this was considerably higher (96 guidelines on average in contrast to 24 questions or 8 tasks on average).
- The total average duration of all evaluation sessions was 76 minutes.
- A lot of extensive sources and technical literature are available to develop own variants of evaluation techniques upon their basis. These sources are listed in Table 3.2.3.
- The literature of Jakob Nielsen—providing information on multiple evaluation techniques—as well as his set of heuristics—specifically intended for heuristic evaluation—were applied rather often, probably due to the fact, that he was among the first researchers to investigate usability evaluation techniques and thus is quite commonly known.
- On the other hand, we found some techniques quite rarely applied. In the case of the walkthrough technique and focus group research the main reason might be the required evaluators; recruiting appropriate expert evalautors—able to conduct walkthroughs—or whole groups of users—willing to discuss the system at a scheduled meeting—can be quite a challenging and/or expensive task. In the case of eye tracking, not only further research on how to conduct such evaluations and analyze the results is required, but also a broader distribution of the required hardware. Concerning guideline-based evaluation and the controlled experiment the main disadvantages assumedly are the evaluators' required expertise and the overall effort.
- Several techniques already established in general software engineering—such as iterative development or pilot testing—also offer potential benefits for usability evaluation. In some cases we found them applied already, but probably they should be used even more extensively in future evaluations.

In summary we found that many techniques for usability evaluation are applicable and are also actually used within the context of applied scientific theses from the field of computer science. Certainly many basic techniques, that can be flexibly adapted to fit a more specific context as well as valuable sources exist for developing one's own tailored evaluation approach.

A Heuristics

Heuristic evaluation—as proposed by Molich and Nielsen in the 1990s—is based on a rather short set of quite generally worded guidelines—the heuristics. Though it is also possible to apply more detailed interface or web design guidelines during a heuristic evaluation, we decided—due to the large amount of existing guidelines—to introduce only well-known heuristics in the narrower sense; that is, short sets (< 20 items) of rather generally worded rules.

For each set a short summary of its development is provided first, followed by the listing of the actual heuristics. Moreover, we found it interesting to compare the different sets and summarized the most interesting findings of the comparisons subsequently to the listing of each set of heuristics.

The 10 heuristics of Nielsen were the first developed specifically for heuristic evaluation. Thus, they constitute one of the most frequently used and established sets of heuristics. Moreover, Nielsen's set has served as a basis for the development of many of the other sets presented here. Therefore we considered Nielsen's heuristics as the basis reference for the comparison of the sets. For each of the other sets, we then primarily investigated the similarities and differences compared to Nielsen's heuristics. Where appropriate, we also compared the different sets among each other.

Therefore, the heuristics—except for Nielsen's set—are annotated in brackets with a short reference to the heuristic(s) they resemble. For example, (**A.1–4**) stands for the fourth heuristic of the heuristics presented in Appendix A.1. Although we primarily drew comparisons between Nielsen's heuristics and the various other sets, we sometimes found that a heuristic was not related to any of Nieslen's heuristics, but rather to one of the heuristics of the other sets. Likewise it occurred, that some heuristics were related as well to Nielsen's set, but at the same time also to other sets. Therefore, references to Nielsen's set—the basis set—are represented in bold face type, references to other sets are printed in normal type. Furthermore, two additional annotations were applied: the annotation *derived*, if a heuristic resembles the referenced heuristic(s) on some points, but contains additional aspects, too; the annotation *new* on the other hand was used whenever a heuristic was introduced newly, that is, when it was neither contained in Nielsen's original set nor in any of the other sets. To provide an example, a heuristic, annotated by (**A.1–5**, A.6–3 / derived), is related in some points to heuristic # 5 of Nielsen's set and to heuristic # 3 of the set in Appendix A.6, but contains additional aspects, asddressed in none of the two referenced heuristics.

Subsequently, 8 sets of heuristics in total are introduced. Appendices A.1 to A.5 present sets, originally developed and intended for heuristic evaluation. In contrast to that, appendices A.6 to A.8 provide sets originally intended to be used as design guidelines. As these guidelines also contain just a small amount of generally worded guidelines, they are also suitable for use in heuristic evaluations.

A.1 Nielsen — The 10 Heuristics

Nielsen's 10 heuristics are one of the most frequently applied set of heuristics today, resulting from several years of research and refinement. In 1990, Nielsen and Molich [99, 115] proposed a first set of 9 heuristics. One more heuristic—*Help and Documentation*, heuristic #10—was added by the authors soon after. A detatiled description of these original 10 heuristics is provided in Nielsen's book "Usability Engineering" [106, p. 115 ff.]. This original set has been further revised by Nielsen [107] in 1994 and since then has undergone no further changes. In the following, the 10 revised heuristics of 1994 are listed in their original wording as published, for example, in the book "Usability Inspection Methods" [108, p. 30] by Nielsen & Mack, or on Nielsen's website [105].

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

2. Match between system and the real world
   The system should speak the users' 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.

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.

4. Consistency and standards
   Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention
   Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall
   Minimize the user's memory load by making 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.

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. Allow users to tailor frequent actions.

8. Aesthetic and minimalistic 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.

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.

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. 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.

A.2 Muller et al. — Heuristics for Participatory Heuristic Evaluation

Muller et al. [102] regarded Nielsen's original heuristics as rather *product-oriented*—that is, focussing on the system as a self-contained unit without adequately taking into account the context of use. Thus they developed three additional heuristics to extend the set of Nielsen in a more *process-oriented* way, especially regarding the context, or more precisely, the system's suitability for the users and their actual work needs. The resulting set of 13 heuristics was published by Muller et al. [102] in 1995. This set consisted of the 10 original heuristics of Molich and Nielsen [106, p. 20], and their three novel ones—*11. Respect the user and his/her skills*, *12. Pleasurable experience with the system*, and *13. Support quality work*.

In the course of further research, Muller et al. developed an adapted variation of the heuristic evaluation, the *participatory heuristic evaluation* (see section 2.3). They also revised their original set of heuristics to better fit the newly developed evaluation technique. As participatory heuristic evaluation incorporates users as evaluators into the evaluation process, the authors also adapted the wording of their heuristics to be easier understandable and applicable by non-expert evaluators. This was achieved through the application of principles of technical-writing and user-oriented documentation. The following list presents the 15 heuristics of 1998 [101] in their original wording, that have not undergone further changes to date.

1. System status (**A.1**–**1**)
   The system keeps users informed about what is going on through appropriate feedback within a reasonable time.

2. Task sequencing (**A.1**–**3** / derived)
   Users can select and sequence tasks (when appropriate), rather than the system taking control of the user's actions. Wizards are available but are optional and under user control.

3. Emergency exits (**A.1**–**3**)
   Users can easily find "emergency exits" if they choose system functions by mistake (emergency exits allow the user to leave the unwanted state without having to go through an extended dialogue. Users can make their own decisions (with clear information and feedback) regarding the costs of exiting current work. They can access undo and redo operations.

4. Flexibility and efficiency of use (**A.1**–**7**, A.5–12, A.7–2)
   Accelerators are available to experts, but are unseen by the novice. Users are able to tailor frequent actions. Alternative means of access and operation are available for users who differ from the "average" user (e.g., in physical or cognitive ability, culture, language, etc.).

5. Match between system and the real world (**A.1**–**2**)
   The system speaks the user's language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Messages are based on the user's real world, making information appear in a natural and logical order.

6. Consistency and standards (**A.1**–**4**)
   Each word, phrase, or image in the design is used consistently, with a single meaning. Each interface object or computer operation is always referred to using the same consistent word, phrase, or image. Follow the conventions of the delivery system or platform.

7. Recognition rather than recall (**A.1**–**6**)
   Objects, actions, and options are visible. The user does not have to remember information from one part of the dialogue to another. Instructions for use of the system are visible or easily retrievable whenever appropriate.

8. Aesthetic and minimalistic design (**A.1**–**8**)
   Dialogs do not contain information that is irrelevant or rarely needed (extra information in a dialog competes with the relevant units of information and diminishes their relative visibility).

9. Help and documentation (**A.1**–**10** / derived)
   The system is intuitive and can be used for the most common tasks without documentation. Where needed, documentation is easy to search, supports a user task, lists concrete steps to be carried out, and is sized appropriately to the user's task. Large documents are supplemented with multiple means of finding their contents (tables of contents, indexes, searches, etc.).

10. Help users recognize, diagnose, and recover from errors (**A.1**–**9**)
    Error messages precisely indicate the problem and constructively suggest a solution. They are expressed in plain (users') language (no codes). Users are not blamed for the errors.

11. Error prevention (**A.1**–**5**)
    Even better than good error messages is a careful design that prevents a problem from occurring in the first place. Users' "errors" are anticipated, and the system treats the "error" as either a valid input or an ambiguous input to be clarified.

12. Skills (new)
    The system supports, extends, supplements, or enhances the user's skills, background knowledge, and expertise. The system does not replace them. Wizards support, extend, or execute decisions made by users.

13. Pleasurable and respectful interaction with the user (new)
    The user's interactions with the system enhance the quality of her or his experience. The user is treated with respect. The design reflects the user's professional role, personal identity, or intention. The design is aesthetically pleasing—with an appropriate balance of artistic as well as functional value.
14. Quality work (new)
    The system supports the user in delivering quality work to her or his clients (if appropriate). Attributes of quality work include timeliness, accuracy, aesthetic appeal, and appropriate levels of completeness.
15. Privacy (new)
    The system helps the user to protect personal or private information—belonging to the user or to his clients.

Basically, most of the heuristics of Muller et al. are quite similar to Nielsen's set as presented in Appendix A.1, only their wording has been adapted in some points. Although heuristic # 2—*Task sequencing*—is derived from Nielsen's third heuristic, the authors explicitly specify the need for enabling the user to control the system, not only concerning erreneous actions, but more generally in every aspect. Muller et al. also slightly modified heuristic # 4—although they named it exactly as its basic heuristic (Nielsen's # 7)—*Flexibility and efficiency of use*—, the authors explicitely mention the need to consider not only users, differing in their level of expertise, but also differing in age, physical or cognitive abilities, culture, or language. This strongly resembles Shneiderman's rule # 2 (see Appendix A.7). Muller et al. also added a further aspect to their ninth heuristic—*Help and documentation*. In contrast to Nielsen, they suggest to supplement large documentations with multiple means supporting the finability of topics. Furthermore, the authors added 4 heuristics (# 12 to # 15) to evaluate the fit of the system to the users and their work needs.

A.3 Constantine & Lockwood—11 Heuristics

Based on their own research over several years, Constantine & Lockwood [25, p. 45-63] provide 11 heuristics in their book "Software for Use". The authors aimed at providing a set of broad, simple, and easy-to-remember rules, that could—in their own opinion—be especially useful for ordinary developers that have to make interface design decisions. Thus, these rules could also serve as a valuable basis for conducting a heuristic evaluation.

1. Access (**A.1–6**, A.4–1, A.5–3 / derived)
   The system should be usable, without help or instruction, by a user who has knowledge and experience in the application domain but no prior experience with the system. Interfaces should by their very organization and construction guide users in how to use them.
2. Efficacy (**A.1–7**)
   The system should not interfere with or impede efficient use by a skilled user who has substantial experience with the system. Features that make usage easy for beginners should not make things harder for more experienced users.
3. Progression (**A.1–7**, A.5—9, A.8—11)
   The system should facilitate continuous advancement in knowledge, skill, and facility and accomodate progressive change in usage as the user gains experience with the system. Good designs should help their users in becoming power users.
4. Support (A.2–14, A.5–1, A.5–2, A.5—11 / derived)
   The system should support the real work that users are trying to accomplish by making it easier, simpler, faster, or more fun or by making new things possible. See software engineering as a chance to reengineer the work iteself.
5. Context (new)
   The system should be suited to the real conditions and actual environment of the operational context within which it will be deployed and used.

6. Structure (**A.1–2**, **A.1–4**, A.6–1, A.8–12 / derived)
   Organize the user interface purposefully, in meaningful and useful ways based on clear, consistent models that are apparent and recognizable to users, puttting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. Use metaphors carefully and only where appropriate.

7. Simplicity (**A.1–2**, A.4–13, A.6–4 / derived)
   Make simple, common tasks simple to do, communicating clearly and simply in the user's own language and providing good shortcuts that are meaningfully related to longer procedures. Tasks that are frequently used and simple for the user should be simple in the interface.

8. Visibility (**A.1–6**, **A.1–8**)
   Keep all needed options and materials for a given task visible, but only those at a time, that are needed to accomplish the given task. Don't distract the user with extraneous or redundant information, or by presenting every possible option at any given time.

9. Feedback (**A.1–1**, **A.1–2**, **A.1–9** / derived)
   Keep users informed of actions or interpretations, changes of state or condition, and errors or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users. Ensure, that the message will be noticed, read, and understood.

10. Tolerance (**A.1–3**, **A.1–5**, **A.1–9** / derived)
    Be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing while also preventing errors wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions reasonably. Assure at least, that software does not do something stupid when confronted with unexpected input or actions.

11. Reuse (**A.1–4**)
    Reuse internal and external components and behaviors, maintaining consistency with purpose rather than merely arbitrary consistency, thus reducing the need for users to rethink and remember. Aim for consistency in appearance, placement, and behaviour. Achieve consistency through reusing existing components.

Nearly all the heuristics of Constantine & Lockwood can be traced back to Nielsen's 10 heuristics. An exception is heuristic # 5, as this specifically addresses the operational context of use of a system. This issue has is not directly addressed by any of the other heuristics. Muller et al. indeed provide heuristics (# 12 to # 15) addressing a system's context, but they focus on user-specific issues as, for example, the user's actual needs or goals. Constantine & Lockwood in contrast address the operational context of the system, that is, the real conditions and actual environment within which the system will be used. Moreover it is remarkable, that the authors—similar to Tognazzini (see Appendix A.8), for example—did not include Nielsen's tenth heuristic *Help and documentation*.

A.4 Kamper — Lead, Follow, and Get Out of the Way

Robert J. Kamper [76] proposed a more recent, revised set of heuristics in 2002. He aimed at providing simple and unified heuristics, that should—similar to metrics—be applicable accross varying technologies and understood in several disciplines to measure the ease of use. Kamper proposes a total of 18 heuristics, that he developed and refined on the basis of Nielsen's 10 heuristics of 1994 (see Appendix A.1). The author established three main principles—*1. Lead*, *2. Follow*, and *3. Get out of the way*—to categorize the heuristics. The main properties of an usable interface, summarized under the *Lead* principle are visibility of the tasks and goal(s) that can be achieved by using the system, simplicity that eases the usage by novice users, and robustness, that enables expert users to obtain goals quick, but with a minimum of errors. The *Follow* principle covers heuristics concerning the support, the system provides to its users for working with the system. The last principle—*Get out of the way*—contains heuristics for evaluating to what extent the system lets its users perform the required actions efficiently. The heuristics were categorized into groups of six under the three principles, as listed below:

A. *Lead the user to successful achievment of goals*

1. Make interface functions obvious and accessible to the user (**A.1–6**)
2. Prevent the possibilities of errors on the part of the user—hide, disable, or confirm inactive or potentially destructive actions (**A.1–5**)
3. Make labels and names distinct from each other—avoid ambiguity and confusion (**A.1–4** / derived)
4. Provide clear, concise prompts in users' own terminology and language (**A.1–2**)
5. Provide safe defaults for inputs—recognition, not recall, of information (**A.1–6** / derived)
6. Support the natural workflow or taskflow of the user (**A.1–2**)

B. *Follow the user's progress and provide support as needed*

7. Provide feedback on all actions (**A.1–1**)
8. Provide progress indicators when appropriate due to length of time elapsed during action (**A.1–1** / derived)
9. Provide error messages that offer solutions to problems (**A.1–9**)
10. Provide feedback on successful completion of a task (**A.1–1**, A.7–4)
11. Provide ability to save input as template in future, record macros, customize preferences, and so forth (**A.1–7** / derived)
12. Provide goal- and task-oriented online help and documentation (**A.1–6**, **A.1–10** / derived)

C. *Get out of the way to allow the user to perform tasks efficiently and effectively*

13. Minimize the number of individual actions needed to perform a task (**A.1–8**, A.6–4, A.3–13)
14. Maintain consistency and adhere to platform conventions and user interface standards (**A.1–4**)
15. Allow the users to maintain control—provide undo, redo, and user exits (**A.1–3**)
16. Provide an aesthetic and minimalistic design—shield user from minutiae unless desired by user (**A.1–8**)
17. Provide for multiple skill and task levels (**A.1–7**)
18. Provide shortcuts (**A.1–7**)

As Kamper developed his principles on the basis of Nielsen's 10 heuristics, it is not surprising that his set relates quite well to Nielsen's original heuristics. The main difference is, that Kamper divided some of Nielsen's original heuristics into two or more individual heuristics for his own set, and only in one case combined two of Nielsen's original heuristics into one of his own set. Thus, in contrast to some of the other presented sets, Kamper's heuristics are somewhat more specific, mostly only focusing on one issue at a time.

A.5 Sarodnick/Brau—Heuristics on the Basis of ISO 9241/10

Sarodnick and Brau [133, p.140] found that most sets of heuristics, including Nielsen's original set, were developed and modified between the 1990s and 2002. Thus those sets of heuristics seem in their opinion somewhat outdated, not (enough) taking into account modern approaches and concepts, such as *Joy of Use*. Therefore, the authors provide an own, adapted set of heuristics, based on the ISO 9241/10, their experiences from research projects, and literature reviews (Sarodnick & Brau [133, pp. 140-141]). Originally, the authors desrcibed their heuristics and associated explanations in german. We translated the heuristics and their associated description for the following listing. The original term for each item is additionally provided in brackets.

1. Task adequacy [Aufgabenangemessenheit] (**A.1–6**, A.2–12, A.8—1)
   All functionalities required for solving the tasks have to be present within the system. They have to be designed in a way, that they support and relieve the user when performing routine tasks.

2. Process adequacy [Prozessangemessenheit] (A.2–12, A.3–5)
   The system should be optimized to enable the solving of actual tasks in typical environments, it should be related to the higher goal of the actual process, and it should be tailored to the qualification and experiences of the real users.

3. Capability of self-description [Selbstbeschreibungsfähigkeit] (**A.1–1**, **A.1–4**)
   The system status should be provided in a consistent and immediate way. The user should be able to choose the level of detail of the system status information.

4. Controllability [Steuerbarkeit] (**A.1–3**, **A.1–7**, A.8–13)
   The user should be able to control the dialog, and should have the possibility to use several input assistances, or to quit the system without data loss.

5. Conformance with users' expectations [Erwartungskonformität] (**A.1–3**)
   The information should conform to system- and platform-specific concepts. For similar tasks, the dialogs should also be similar, and should be displayed at their expected position.

6. Error tolerance [Fehlertoleranz] (**A.1–5**, **A.1–9**)
   Error messages should be expressed clearly. They should contain, for example, information about the type and the context of the error. The user should be informed about irreversible actions.

7. System- and data-safety [System- und Datensicherheit] (A.8–13)
   The system should always work stable and without data loss, even if users provide defective inputs, or under higher load.

8. Individual adjustability [Individualisierbarkeit] (**A.1–7**, A.4–11)
   The dialog system should be individually adjustable, conforming to the users' preferences, as long as it serves the users' effectiveness, efficiency and satisfaction and does not contradict the required technical or security-related constraints.

9. Learning conductiveness [Lernförderlichkeit] (**A.1–7**, A.3–3, A.8–7, A.8–11)
   Learning approaches, as for example *Learning by Doing*, should be supported through stepwise instructions or navigation aids.

10. Control through perception [Wahrnehmungssteuerung] (**A.1–8**)
    Interface layout should be minimalistic. Groupings, colors, and useful reduction of information, or similar, should be used in a way that the users' attention is directed to the relevant information.

11. Joy of use [Joy of Use] (**A.1–4**, A.3–4, A.8–12)
    Task sequences and graphical design of the system should avoid monotony and appear up to date, but also consider the necessary consistency. Metaphors should be used adequately and should match the context of usage.

12. Intercultural aspects [Interkulturelle Aspekte] (**A.1–7**, A.2–4 A.7–2 / derived)
    The system should be matching a defined user population in terms of, for example, their functional, organizational, or national culture.

Generally, the heuristics of Sarodnick & Brau also match Nielsen's original heuristics quite well. Yet, there are two heuristics within Nielsen's set, that were not addressed here: heuristic # 2 (*Match between the system and the real world*), and heuristic # 10 (*Help and Documentation*). In turn, Sarodnick & Brau focussed more on also incorporating the context of use of the system (see heuristics # 1 and # 2) and the aspect *joy of use*.

A.6 Donald A. Norman—Design principles

In his book "The Design of Everyday Things", Donald A. Norman [117] proposes a set of general guidelines for improving the design of things. Though he summarizes the main principles within the last chapter [117, p. 188 ff.] he mentiones some additional prinicples throughout his book. In the following we present the design principles found along with a summarization.

1. Provide a good conceptual model (**A.1–6**, **A.1–10**, A.8–12 / derived)
   Don't force the user to keep all required knowledge in the head, but make knowledge externally accessible. Provide an appropriate model of the system, so the user is able to predict the effects of his actions. Also provide understandable instructions and/or a system manual.

2. Consistency (**A.1–4**)
   Be consistent in presenting actions and results. Provide a consistent system image. Provide consistency between the system and the users' goals, expectations, and intentions.

3. Feedback (**A.1–1**, A.7–3 / derived)
   Provide information what action has been done and what results have been accomplished. Feedback may be visual, textual, auditory, or tactile. Feedback has to be provided immediately after an action has taken place. Provide up to date information about the current system state.

4. Simplicity (**A.1–6**, **A.1–8**, A.4–13, A.3–7)
   Tasks should have a simple structure, minimizing the required planning or problem solving. Consider limitations of the short-term memory, of the long-term memory, and of attention. Complex tasks should be simplified by restructuring: provide mental aids, make the invisible visible, automate, or change the nature of the task. Be careful with *creeping featurism*—adding new features beyond all reason: either completely avoid it, or organize features, for example, through modularization.

5. Visibility (**A.1–1**, **A.1–6** / derived)
   Users should be able to judge the current state of the system, possible actions, and their effects simply by looking.

6. Natural mapping (**A.1–2**)
   Provide reasonable mappings between controls and their effects. Exploit natural mappings, utilize physical analogies and/or cultural standards.

7. Use constraints and affordances (**A.1–5**, **A.1–6** / derived)
   Use natural and artificial constraints, to limit the users actions to the currently reasonable ones. Constraints might be phyisical, semantic, cultural, or logical. If required, use forcing functions to assure certain behaviour or actions. Use affordances to hint to possible use, functions, actions; suggest a range of possibilities.

8. Design for error (**A.1–3**, **A.1–5**, **A.1–9**)
   Assume that any error can and will be made. Plan for errors: allow the user to recover from errors and to understand what caused the error. Make operations reversable, especially any unwantend outcome. Provide undo and redo of actions to design explorable systems. Don't blame the users for making errors.

All of Norman's design principles—as the annotations show—can either be directly derived, or are composed of several of Nielsen's heuristics. Remarkably is, that Norman's set does not include any principle resembling Nielsen's seventh heuristic—*Flexibility and efficiency of use*.

A.7 Shneiderman — The Eight Golden Rules

Another established set of interface design guidelines—the *eight golden rules*—was developed by Ben Shneiderman [139, pp. 74-75] over several years. He proposes these rules in his book "Designing the User Interface", along with a detailed explanation for each item. Those principles are intended to be applicable in most interactive systems—for example, desktop applications, web

applications and -sites, or mobile applications—so they are quite generally worded. This in turn makes them suitable to be applied during a heuristic evaluation of interfaces. In the following we provide the eight golden rules along with a summarization of Shneiderman's original explanation of each rule.

1. Strive for consistency (**A.1–4**, A.8—4)
   Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; consistent color, layout, capitalization, fonts, and others should be employed throughout. Exceptions should be comprehensible and limited in number.

2. Cater to universal usability (**A.1–7**, A.2–4, A.5–12)
   Recognize the needs of diverse users and design for plasticity, faciliating transformation of content. Consider novice-expert differences, age ranges, disabilities, and technology diversity. Add features for novices—for example explanations—, and features for experts—for example shortcuts or faster pacing.

3. Offer informative feedback (**A.1–1**, A.6–3)
   Provide system feedback for every user action. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Consider the visual presentation of showing changes of the objects of interest.

4. Design dialogs to yield closure (**A.1–1**, **A.1–2** / derived)
   Sequences of actions should be organized into groups with a beginning, middle, and end. Provide informative feedback at the completion of a group of actions.

5. Prevent errors (**A.1–5**, **A.1–9**)
   Design the system such that users cannot make serious errors. In case of an error, the system should offer simple, constructive, and specific actions for recovery. Erroneous actions should leave the system state unchanged, or instructions about restoring the state should be provided.

6. Permit easy reversal of actions (**A.1–3**)
   As much as possible, actions should be reversible to encourage users' exploration of unknown options. Units of reversibility could be single actions, a data-entry task, or a complete group of actions, such as entry of a name and address block.

7. Support internal locus of control (**A.1–3**, **A.1–6** / derived)
   Users should feel that they are in charge of the interface and that it responds to their actions. Avoid surprising interface actions, tedious data-entry sequences, difficulties in finding necessary information, and inability to produce the action desired. Make users initiators of actions rather than just responders.

8. Reduce short-term memory load (**A.1–6**, **A.1–8**, **A.1–10** / derived)
   Human information processing is constrained—7 plus/minus 2 chunks of information are believed to be rememberable. Therefore keep the interface simple and multiple-page displays consolidated. Also reduce window-motion frequency and allow for sufficient training time for codes, mnemonics, and sequences of actions. Where appropriate, provide online access to command-syntax forms, abbreviations, codes, and other necessary information.

Shneiderman's golden rules relate quite well to Nielsen's 10 heuristics. As the annotations show, each of the rules can be somehow traced back to one (or several) of Nielsen's original heuristics. In most cases Shneiderman's rules differ from Nielsen's heuristics in the level of detail or the explanations. For example, in his first rule—*Strive for consistency*—Shneiderman explicitly names examples of interface components that should be kept consistent, similar to Tognazzini (see Appendix A.8). Shneiderman's fourth rule also concerns one aspect, not directly adressed within the other sets of heuristics: designing dialogs with a clearly identifiable beginning, middle, and end, and provide adequate feedback on the completion of such sequences of actions.

A.8 Tognazzini—First Principles of Interaction Design

Bruce Tognazzini [151] developed his *First Principles of Interaction Design* in 2003. The principles are intended to be fundamental for designing interfaces of desktop applications as well as for the web. Tognazzini himself summarizes his principles as follows:

> Effective interfaces are visually apparent and forgiving, instilling in their users a sense of control. Users quickly see the breadth of their options, grasp how to achieve their goals, and do their work. Effective interfaces do not concern the user with the inner workings of the system. Work is carefully and continuously saved, with full option for the user to undo any activity at any time. Effective applications and services perform a maximum of work, while requiring a minimum of information from users. (Website of Bruce Tognazzini [151])

Compared to several other sets of heuristics, the description of some of the principles is rather detailed. Nonetheless Tognazzini's principles still might be used as a valuable basis for a heuristic evaluation. In the following we listed the principles along with a summary of their description as presented on Tognazzini's website [151].

1. Anticipation (**A.1**–**6**, A.5—5)
   Anticipate the user's wants and needs. Do not expect users to search for or gather information or evoke necessary tools. Bring to the user all the information and tools needed for each step of the process.

2. Autonomy (**A.1**–**1**, **A.1**–**3** / derived)
   Provide user-autonomy but don't abandon all rules. Keep users aware and informed with status mechanisms (helps them to feel them in control of the system). Keep status information up to date and within easy view.

3. Color Blindness (new)
   As still many people suffer from color blindness in one or another form, never use color exclusively to convey information. Provide clear, secondary cues, that can consist of anything from the subtlety of gray scale differentiation to having a different graphic or different text label associated with each color presented.

4. Consistency (**A.1**–**4**, A.7—1)
   Cater for a consistent appearance of those small visible structures—icons, size boxes, scroll arrows, and the like. Location is only just slightly less important than appearance, thus standardize location, where it makes sense. Make objects consistent with their behaviour, but avoid uniformity. Make objects look differently if they act differently, but be visually consistent if they act the same. The most important consistency is consistency with user expectations. Ascertain user expectations through testing.

5. Defaults (A.4–5)
   Defaults should be appropriate, but also easy recognizable and replaceable—for example they should be preselected for a quick recognition and replacement. Do not use the actual word *default* in an application or service, but consider more specific terms as, for example, *standard*, *use customary settings*, or similar.

6. Efficiency of the user (**A.1**–**7**, **A.1**–**9** / derived)
   Look at the user's productivity, not the computer's. Keep the user occupied—avoid long system response times. Write help messages tightly and make them responsive to the problem; this eases comprehension and efficiency. Menu and button labels should have the key word(s) first.

7. Explorable Interfaces (**A.1**–**3**, **A.1**–**7** / derived)
   Provide clear landmarks, but enable users to explore the interface. Provide more than one way to solve a task to support those, who just want to get the job done quickly as well as those, who wolud like to more deeply explore the system. Moreover provide clear directives for novice users. Stable perceptive clues enable users to feel comfortable and familiar with the system. Allways allow for reversible actions; provide undo instead of several confirmation dialogues, and always provide an escape action, so users never feel trapped.

8. Fitt's Law (new)
   According to the *Fitt's Law*, the time to acquire a target is a function of the distance to and size of the target. Make objects' sizes proportional to their importance within the interface and place them within appropriate reach.

9. Human interface objects (**A.1**–**2**)
   Design interface objects for mapping to the real world of the user—examples are *folders* or the *trashcan*. Design using metaphors. Human-interface objects have a standard way of interacting, standard resulting behaviors, and should be understandable, self-consistent, and stable.

10. Latency Reduction (**A.1**–**1**, **A.1**–**8** / derived)
    Make use of multithreading. Make the interface faster—eliminate all elements of the application that are not helping. Reduce the users' experience of latency: provide visual feedback to button clicks within considerable time, provide progress bars and messages, display animated hourglasses for actions lasting 0.5 to 2.0 seconds, return noticably from lengthy actions in providing beeps, or large visual displays.

11. Learnability (**A.1**–**7**, A.5—9, A.3—3 / derived)
    Limit the trade-offs between learnability and usability—learnability mostly focusses on easing the use of the system for novice users, but usability in general might also include advanced features for experienced users.

12. Metaphors (**A.1**–**1**, A.3—6, A.6—1)
    Chose metaphors well. Chose those that will enable users to instantly grasp the finest details of the conceptual model. Create stories and visible pictures in the users' minds.

13. Protect users' work (**A.1**–**5** / derived)
    Ensure that users never lose their work as a result of error on their part, the vagaries of Internet transmission, or any other reason other than the completely unavoidable, such as sudden loss of power to the client computer. Provide automatic data-saving mechanisms.

14. Readability (new)
    Use high contrasts and appropriate font sizes for displaying texts. Particularly pay attention to the needs of elder people.

15. Track State (new)
    Track the steps of the user, that is, where in the system he has been and where he left off in the last session. Moreover it might in some cases also be interesting, what the user has done. Enable users to pause their work and continue at any time from exactly the point, where they left.

16. Visible Navigation (**A.1**–**2**, **A.1**–**6** / derived)
    Avoid invisible navigation as most users cannot and will not want to build elaborate mental maps, or get tired when they have to. Reduce navigation to a minimum, keep it clear and natural. Provide maps and/or other navigational aids to offer users a greater sense of mastery and autonomy. Especially web navigation often is invisble; therefore add the layers of capability and protection that users want and need.

As the annotations show, most of Tognazzini's principles are closely related to Nielsen's 10 heuristics. Exceptions are the principles # 3, # 8, # 14, and # 15. The three former describe quite specific design guidelines, concerning the issues *usage of colors*, *placement of objects within an interface*, and *textual design*. Such detailed design suggestions were not included in the sets presented up to this point. Principle # 15 constitutes a new issue, not directly addressed by any other heuristic before: tracking the users' steps and the system state. This could be used to enable the users to start working exactly at the point he left the system the last time, without, for example, having to open several files they recently worked with by themselfes. On the other hand, tracking could support later evaluation of the actual usage of the system. Remarkable is also, that—similar to Constantine & Lockwood—Tognazzini did not include any principle, based on Nielsen's tenth heuristic *Help and documentation*.

B Usability Metrics

In the following, a collection of the most frequently applied usability metrics is presented. The listing consists of metrics described by the following authors: Nielsen [106, p. 194-195], Stone et al. [143], Constantine & Lockwood [25, pp. 454 ff.], Bevan & Macleod [10], Rubin & Chisnell [132, pp. 166, 249 ff.] and Tullis & Albert [152].

Some of the basic metrics—for example task duration—are measured plainly in numbers or in an amount of time, resulting in data sets with a value for each user. There exist four commonly used techniques to interpret the values of such data sets:

- The ***Arithmetic Mean*** $= \frac{Sum\ of\ All\ Participants'\ Values}{Number\ of\ Participants}$
  The arithmetic mean is a common technique for caluculating an average value out of a set of values. In the case of measuring user performance, this is an easy method to determine the average performance of the whole group of participants.
- The ***Median***
  The median is the middle value that separates the higher half and the lower half of the data set from each other. To determine the median, the values of the data set have to be listed in ascending order. The median then is the value located exactly in the middle of the set. If the number of values was even, there remain two middle values. In this case, the median is the mean of these two values.

  The median is the appropriate means to calculate an avarage of an dataset, if its values are very skewed either left or right—that is, if the highest and lowest values are very different from all other values. Therefore it is especially suitable for calculating some kind of average performance of a group of participants, when this group included some participants that performed exceedingly good or bad.
- The ***Range (high/low)*** of values
  The range of values is defined through the lowest and the highest value. If—in the case of performance measurement—this range is exceptionally high, one should further investigate the potential reasons. For example, one should ask, why some participants obviously performed much better or much worse than the others, and whether this is due to one participant's personal skills, or whether this might be representative of the majority of future users.
- The ***Standard Derivation (SD)***
  The SD indicates, to what degree the examined values differ from each other, that is, how closely the values are clustered around the mean value. The basic formula for calculating the SD is

  $$SD = \frac{\sqrt{\sum x^2 - \frac{(\sum x)^2}{n}}}{n-1}$$

  with $\sum x^2$ = sum of the squares of each of the values and $\sum x$ = sum of all values

  The SD can be applied, if one wants to investigate, whether a group of users performed quite homogeneous or rather dissimilar. If the latter is the case, one should again investigate the potential reasons.

Usability Metrics

1. ***Success Rate / Correctness*** (Percentage of tasks that users complete correctly)
   According to Nielsen [105, Alertbox Feb. 18, 2001] the fundamental metric, because if users can't accomplish their tasks, all other measures are irrelevant. Also referred to as *correctness*, e.g. [25, pp. 454 ff.].
   $Success\ Rate\ = \frac{Correctly\ Completed\ Tasks}{Total\ Number\ of\ Tasks} \times 100$
2. ***Task Duration*** (The time a task requires to be accomplished [132, pp. 250 ff.])
3. ***Task Accuracy*** (The number of participants that performed successfully [132, pp. 250 ff.])
   Accuracy $= \frac{Number\ of\ Successful\ Participants}{Total\ Number\ of\ Participants}$ (basic formula)

   Rubin & Chisnell describe three variations how to calculate the accuracy, differing only in the participants, included in the calculation:

   1. ***Participants performing successfully but with assistance***

   Includes all participants that somehow managed to accomplish the task, even if they needed assistance or

exceeded a given benchmark time limit.

2. ***Participants performing successfully***

Includes all participants that somehow managed to accomplish the task successfully on their own—even if they exceeded a given benchmark time limit.

3. ***Participants performing successfully within time***

Includes participants that not only accomplished the given task successfully on their own, but also within a predefined benchmark time limit.

4. ***Completeness*** (Percent of total assigned tasks completed in allotted benchmark time [10])
   Completeness = $\frac{Tasks\ Completed}{Number\ of\ Tasks} \times 100$

5. ***Effectiveness*** (Correctly completed tasks as a percentage of total number of tasks [10])
   Effectiveness = $\frac{1}{100} \times (Completeness \times Correctness)$

6. ***Efficiency*** *(Effectiveness per unit of time, effort, or total cost)*
   Bevan & Macleod [10] distinguish three possible measures of efficiency:
   1. Temporal Efficiency = $\frac{Task\ Effectiveness}{Task\ Time}$
   2. Human Efficiency = $\frac{Task\ Effectiveness}{Effort}$ Effort can be derived from workload measures (see item 22)
   3. Economic Efficiency = $\frac{Task\ Effectiveness}{Total\ Cost}$ Total Cost conforms to the resources consumed for the task

   According to Bevan & Macleod [10], efficiency can only reasonably be judged within a proper context, that is, those values only have meaning, when compared against efficiency benchmarks. Thus, efficiency measures can be used for comparing...

   a. ... two or more similar products or different versions of one product when used by the same user group in the same envorinment for the same tasks

   b. ... two or more types of users when using the same product for the same tasks in the same environment

   c. ... two or more tasks when carried out by the same users on the same product in the same environment

7. ***Dead Time*** (Time, when the user is not interacting with the system [106, p. 194])
   Nielsen distinguishes two variations of dead time, that should be approached–and therefore also measured—seperate from each other:
   1. response-time delays (the user waits for the system)
   2. thinking-time delays (the system waits for the user to perform the next actions)

   Bevan & Macleod [10] define a metric similar to the *thinking-time delays*—***Unproductive Time***. In contrast to Nielsen, they specify unproductive time more detailed as consisting of periods during which users seek help (Help Time), search hidden structures (Search Time), and try to overcome problems (Snag Time).

8. ***Productiveness*** (Percent of time spent productively [25, pp. 454 ff.])
   Also referred to as *Productive Period (PP)* [10].
   PP = $\frac{Task\ Time - Unproductive\ Time}{Task\ Time} \times 100$

9. ***Error Rate*** (Ratio between successful tasks and failures [105, 106])
   Error Rate = $\frac{Successful\ Tasks}{Failed\ Tasks}$

10. ***Number of Errors*** (Number of errors made by the user [106, p. 194])

11. ***Recovering Time*** (The time users need to recover from errors [106, p. 194])

12. ***Help System Usage*** (Extent to which the users make use of the help system)
    Nielsen [106, p. 194] suggests not only to measure the number of times, the help system is used, but also the duration of the help system usage

13. ***Expert Help*** (The number and/or type of hints or prompts the user requires [132, p. 166])

14. ***Used Commands/Features*** (Number of commands/features the user actually utilized)
    Nielsen [106, p. 194] distinguishes two possible measures: the *absolute* number of commands/features used, and the number of *different* commands/features used

15. ***Unused Commands/Features*** (The number of commands/features that were never used [106, p. 194])

16. ***Recallable Features*** (The number of features the user remembers after the test [106, p. 194])

17. ***Critical Statements Ratio (CSR)*** (Proportion of positive towards critical statements)
    The number of distinct positive towards critical statements concerning the system [106, p. 194] are collected during the test. Both numbers can serve as metrics on their own, their combination results in the CSR:
    CSR = $\frac{Positive\ User\ Statements}{Critical\ User\ Statements}$

18. ***Sidetracking*** (Extent to which the user is sidetracked from focusing on the task)
    Interesting is not only the plain number of times, the user is sidetracked, but also the context [106, p. 194] (the actual used feature or the task at hand).

19. ***Training Time*** (The time it takes until users achieve specific benchmark measures in performing the tasks [132, p. 166])

20. ***Learning Rate*** (Rate at which users learn to use the system)
    According to Bevan & Macleod [10], the learning rate can be assessed in two ways:
    1. measure the rate of increase in specified metrics when the user repeats evaluation sessions
    2. measure the efficiency of a particular user relative to an expert:
    Relative User Efficiency = $\frac{User\ Efficiency}{Expert\ Efficiency} \times 100$

21. ***Subjective Satisfaction*** (The subjective satisfaction of the users [10])
    To gain an insight of the subjective satisfaction of the users, it is advisable to let them answer short questionnaires or scales, covering the usefulness of the product, the users' satisfaction with functions and features, a rating whether users or technology had the control during usage, and the users' perception, whether the task is appropriately supported by the system

22. ***Cognitive Workload*** (Mental effort required to perform a task)
    Bevan & Macleod [10] differntiate two kinds of measures:
    1. **Heart Rate Variability** as an objective measure; as people invest mental effort, the heart rate variability has showed to be reduced.
    2. **Questionnaires** as subjective measure; the mental workload of users can be assesed, for example, by the SMEQ (Subjective Mental Effort Questionnaire) or the NASA TLX (NASA Task Load Index)

Apart from these metrics that are calculated on the basis of user performance or satisfaction measurement, Constantine & Lockwood [25, pp. 426 ff.] suggest a suite of 5 more elaborate metrics for the evaluation of interface usability. Calculated on the basis of use cases and the interface itself those metrics are: *Essential Efficiency, Task Concordance, Task Visibility, Layout Uniformity, and Visual Coherence.* Constantine & Lockwood provide more detailed information on their calculation and usage in their book.

References

[1] Apple Computer Inc, Apple Web Design Guide, retrieved July 15, 2008 from http://www.usability.ru/sources/AppleWeb.pdf.

[2] Apple Computer Inc., Apple Human Interface Guidelines: The Apple Desktop Interface, Addison-Wesley, 1987.

[3] Apple Computer Inc., Macintosh Human Interface Guidelines, Addison-Wesley Professional, 1993.

[4] Apple Computer Inc., Newton 2.0 User Interface Guidelines, Addison Wesley Longman, 1996.

[5] A. Baker, Better Flash Websites, Retrieved July 15, 2008 from http://www.merges.net/theory/20010416.html.

[6] J. J. Baroudi, W. J. Orlikowski, A Short-Form Measure of User Information Satisfaction: A Psychometric Evaluation and Notes on Use., Journal of Management Information Systems 4 (4) (1988) 44–59.

[7] T. Bartel, Die Verbesserung der Usability von Websites auf der Basis von Web Styleguides, Usability Testing und Logfile-Analysen [Enhancing Website Usability on the Basis of Web Styleguides, Usability Testing, and Logfile Analysis], Master's thesis, University of Hildesheim, Germany (2001).

[8] J. M. C. Bastien, D. L. Scapin, Ergonomic Criteria for the Evaluation of Human-Computer Interfaces, Tech. Rep. 156, INRIA (June 1993).

[9] C. L. Beard, An Internet Search Interface for the Ackland Art Museum Collection Database, Master's thesis, University of North Carolina at Chapel Hill, USA (2004).

[10] N. Bevan, M. Macleod, Usability Measurement in Context, Behaviour and Information Technology 13 (1994) 132–145.

[11] N. Bevan, L. Spinhof, Are Guidelines and Standards for Web Usability Comprehensive?, in: Proceedings Part I of the 12th International Conference, HCI International, Beijing, China, 2007.

[12] R. G. Bias, The Pluralistic Usability Walkthrough: Coordinated Empathies, chap. 3, in: Nielsen and Mack [114], pp. 63–76.

[13] R. Blaser, Einsatz und Evaluierung eines evolutionären IT-Konzepts für ein integriertes klinisches Informationssystem [Application and Evaluation of an Integrated Clinical Information System], Ph.D. thesis, Philipps-University of Marburg, Germany (2007).

[14] J. A. Borges, I. Morales, N. J. Rodríguez, Page Design Guidelines Developed Through Usability Testing, chap. 11, in: Forsythe et al. [43].

[15] J. Brooke, SUS: A Quick and Dirty Usability Scale, chap. 21, in: Jordan et al. [73], pp. 189–194.

[16] M. D. Byrne, Cognitive Architecture, chap. 5, in: Jacko and Sears [69], pp. 97–117.

[17] T. Büring, Zoomable User Interfaces on Small Screens—Presentation and Interaction Design for Pen-Operated Mobile Devices, Ph.D. thesis, University of Konstanz, Germany (2007).

[18] S. Card, T. P. Moran, A. Newell, The Keystroke-Level Model for User Performance with Interactive Systems, Communications of the ACM 23 (1980) 396–410.

[19] S. Card, T. P. Moran, A. Newell, The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, 1983.

[20] Y. Chen, L. Huang, L. Li, Q. Luo, Y. Wang, J. Xu, The Experimental Approaches of Assessing the Consistency of User Interfaces, in: Proceedings Part I of the 12th International Conference, HCI International, Beijing, China, 2007.

[21] J. P. Chin, V. A. Diehl, K. L. Norman, Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface, in: CHI '88: Proceedings of the SIGCHI conference on Human Factors in computing systems, Washington DC, USA, 1988.

[22] N. Claridge, J. Kirakowski, Website of the WAMMI project, Accessed July 16, 2008 at http://www.wammi.com/.

[23] G. Cockton, D. Lavery, A. Woolrych, Inspection-Based Evaluations, chap. 57, in: Jacko and Sears [69], pp. 1118–1138.

[24] L. L. Constantine, Devilish Details: Best Practices in Web Design, in: L. L. Constantine (ed.), Proceedings of the First International Conference on Usage-Centered, Task-Centered, and Performance-Centered Design for USE, Ampersand Press, 2002.

[25] L. L. Constantine, L. A. D. Lockwood, Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design, Addison-Wesley Professional, 1999.

[26] L. L. Constantine, L. A. D. Lockwood, Instructive Interaction: Making Innovative Interfaces Self-Teaching, Retrieved July 16, 2008 from http://www.foruse.com/articles/instructive.pdf.

[27] A. Cooper, About Face 3 - The Essentials of Interaction Design, Wiley Publishing Inc., 2007.

[28] DATech (Deutsche Akkreditierungsstelle Technik GmbH), DATech - Leitfaden Usability v 1.1 [DATech—Usability Compendium v 1.1], 2008, Retrieved July 15, 2008 from http://www.datech.de/index.php?id=0030&kat=1.

[29] M. de Jong, T. van der Geest, Characterizing Web Heuristics, Technical Communication 47 (2000) 311–326.

[30] J. A. M. de Nascimento, Usabilidade no Contexto de Gestores, Desenvolvedores e Usuários do Website da Biblioteca Central da Universidade de Brasília [Usability in the Context of Managers, Developers, and Users of the Website of the Central Library at the University of Brasilia], Master's thesis, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Brazil (2006).

[31] D. Decker, Eine vergleichende Analyse der Websites von Anbietern pneumatischer Automatisierungskomponenten—heuristische Usability-Evaluation und zielbasierte Content-Analyse [A Comparative Analysis of Websites of Pneumatic Automation Component Suppliers—Heuristic Usability Analysis and Goal-Based Content Analysis], Master's thesis, College of Higher Education of Stuttgart, Germany (2002).

[32] C. A. Dias, Métodos de Avaliação de Usabilidade no Contexto de Portais Corporativos: Um Estudo de Caso no Senado Federal [Methods for Usability Evaluation in the Context of Corporate Portals: a Case Study in the Senate], Ph.D. thesis, Universidade de Brasilia (2001).

[33] P. Díaz, M. Ángel Sicilia, I. Aedo, Evaluation of Hypermedia Educational Systems: Criteria and Imperfect Measures, in: ICCE '02: Proceedings of the International Conference on Computers in Education, Auckland, New Zealand, 2002.

[34] DIN (Deutsches Institut für Normung e.V.), DIN EN ISO 9241 [17 norms], Available from Beuth-Verlag. Accessed July 18, 2008 at http://www.beuth.de/.

[35] A. Dix, J. Finlay, G. D. Abowd, R. Beale, Human-Computer Interaction, Pearson Education Limited, 2004.

[36] L. L. Downey, Group Usability Testing: Evolution in Usability Techniques, Journal of Usability Studies 2 (2007) 133–144.

[37] A. Drescher, Evaluation des Lernerfolges einer Blended Learning Maßnahme unter Berücksichtigung der Barrierefreiheit [Evaluating the Learning Success of a Blended-Learning Method, Considering Accessibility], Master's thesis, Technical and Economical University of Dresden, Germany (2007).

[38] T. M. Duffy, J. E. Palmer, B. Mehlenbacher, Online Help, Design and Evaluation, Ablex Publishing Corp., 1992.

[39] J. S. Dumas, User-Based Evaluations, chap. 56, in: Jacko and Sears [69], pp. 1093–1117.

[40] J. S. Dumas, J. C. Redish, A Practical Guide to Usability Testing, Intellect Books, 1999.

[41] K. Eilers, F. Nachreiner, K. Hänecke, Entwicklung und Überprüfung einer Skala zur Erfassung subjektiv erlebter Anstrengung [Development and Verification of a Scale for Measuring Subjective Workload], Zeitschrift für Arbeitswissenschaft 40 (1986) 215–224.

[42] J. Fleming, Web Navigation: Designing the User Experience, O'Reilly, 1998.

[43] C. Forsythe, E. Grose, J. Ratner (eds.), Human Factors and Web Development, CRC, 1997.

[44] R. Fujioka, R. Tanimoto, Y. Kawai, H. Okada, Tool for Detecting Webpage Usability Problems from Mouse Click Coordinate Logs, in: Proceedings Part I of the 12th International Conference, HCI International, Beijing, China, 2007.

[45] R. Fukuda, Ergonomische Gestaltung der Webauftritte: Analyse des menschlichen Verhaltens bei der Webnutzung und darauf basierende nutzerspezifische Vorschläge [Designing Ergonomics into Web Presences: Analyzing Human Behaviour while Using the Web and User-Specific Design Suggestions Based on the Analysis], Ph.D. thesis, Technical University of Munic, Germany (2004).

[46] W. O. Galitz, The Essential Guide to User Interface Design—An Introduction to GUI Design Principles and Techniques, Wiley Publishing Inc., 2007.

[47] G. Gediga, K.-C. Hamborg, IsoMetrics: Ein Verfahren zur Evaluation von Software nach ISO 9241/10 [IsoMetrics: an Approach for Software Evaluation in Terms of the ISO 9241/10], chap. 7, in: Holling and Gediga [60], pp. 195–234.

[48] J. Geißler, Design und Implementierung einer stiftzentrierten Benutzungsoberfläche [Design and Implementation of a Pen-Based User Interface], Ph.D. thesis, Technical University of Darmstadt, Germany (2001).

[49] K.-C. Hamborg, IsoMetrics Project Homepage, Accessed July 15, 2008 at http://www.isometrics.uni-osnabrueck.de/.

[50] P. A. Hancock, N. Meshkati (eds.), Human Mental Workload, Elsevier Science, 1988.

[51] S. Hart, L. Staveland, Development of NASA TLX (Task Load Index): Results of Empirical and Theoretical Research, in: Hancock and Meshkati [50], pp. 139–183.

[52] H. R. Hartson, J. C. Castillo, Remote Evaluation for Post-Deployment Usability Improvement, in: Proceedings of AVI '98, Advanced Visual Interfaces, L'Aquila, Italy, 1998, pp. 22–29.

[53] R. Hartwig, Ergonomie multimedialer interaktiver Lehr- und Lernsysteme [Ergonomics of Multimedial, Interactive Teaching and Learning Applications], Ph.D. thesis, University of Lübeck, Germany (2005).

[54] M. Hassenzahl, M. Burmester, F. Koller, AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität [AttrakDiff: a Questionnaire for Measuring Perceived Hedonic and Pragmatic Quality], Mensch & Computer (2003) 87–196.

[55] M. Hassenzahl, A. Platz, M. Burmester, K. Lehner, Hedonic and Ergonomic Quality Aspects Determine a Software's Appeal, in: CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, New York, NY, USA, 2000, pp. 201–208.

[56] J. O. Hauglid, User Interfaces for Accessing Information in Digital Repositories, Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway (2004).

[57] K. Hauser, Entwicklung einer Methode und Pilotstudie zur Langzeitevaluation von adaptiven User Interface Elementen [Developing an Approach for Long-Term Evaluation of Adaptive User Interface Elements, and Pilot Study], Master's thesis, College of Higher Education of Stuttgart, Germany (2004).

[58] M. Hegner, Methoden zur Evaluation von Software [Software Evaluation Techniques], Arbeitsbericht 29, GESIS: InformationsZentrum-Sozialwissenschaften, Bonn (2003).

[59] N. Heinze, Nutzen und Nutzbarkeit des Felsinformationssystems des DAV – eine Usability Studie [Use and Usability of the Mountain Information System of the DAV—a Usability Study], Ph.D. thesis, University of Augsburg, Germany (2007).

[60] H. Holling, G. Gediga, Evaluationsforschung [Evaluation Research], Hogrefe, 1999.

[61] T. Hollingsed, D. G. Novick, Usability Inspection Methods after 15 Years of Research and Practice, in: SIGDOC '07: Proceedings of the 25th annual ACM international conference on Design of communication, El Paso, Texas, 2007, pp. 249–255.

[62] W. Horn, Leistungsprüfsystem [Performance Test System], Hogrefe Verlag für Psychologie, 1983.

[63] IBM, Object-Oriented Interface Design: IBM Common User Access Guidelines, Prentice Hall, 1993.

[64] IBM, IBM Web Guidelines, Retrieved July 15, 2008 from http://interface.free.fr/Interface/ergonomie.html (2000).

[65] IS & T, Usability Guidelines, MIT (Massachusetts Institute of Technology), Retrieved July 15, 2008 from http://web.mit.edu/ist/usability/usability-guidelines.html.

[66] ISO - International Organization for Standardization, Website of the International Organization for Standardization, Accessed June 28, 2008 at http://www.iso.org/iso/home.htm.

[67] B. Ives, M. H. Olson, J. J. Baroudi, The Measurement of User Information Satisfaction, Communications of the ACM 26 (10) (1983) 785–793.

[68] M. Y. Ivory, An Empirical Foundation for Automated Web Interface Analysis, Ph.D. thesis, University of California at Berkeley, USA (2001).

[69] J. A. Jacko, A. Sears (eds.), The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Lawrence Erlbaum Associates, 2003.

[70] B. E. John, D. E. Kieras, The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast, ACM Transactions on Computer-Human Interaction (ToCHI) 3 (4) (1996) 320–351.

[71] B. E. John, D. E. Kieras, Using GOMS for User Interface Design and Evaluation: Which Technique?, ACM Transactions on Computer-Human Interaction (ToCHI) 3 (4) (1996) 287–319.

[72] J. Johnson, GUI Bloopers 2.0, Common User Interface Design Don't and Do's, Morgan Kaufmann Publishers, 2008.

[73] P. W. Jordan, B. Thomas, I. L. McClelland, B. Weerdmeester (eds.), Usability Evaluation in Industry, CRC, 1996.

[74] J. Kaasalainen, User Interface Design and Usability Testing of a Podcast Interface, Ph.D. thesis, Helsinki University of Technology, Finland (2007).

[75] M. J. Kahn, A. Prail, Formal Usability Inspections, in: Nielsen and Mack [114], pp. 141–172.

[76] R. J. Kamper, Extending the Usability of Heuristics for Design and Evaluation: Lead, Follow, and Get Out of the Way, International Journal of Human-Computer Interaction 14 (3&4) (2002) 447–462.

[77] M. Karlsson, Usability from Two Perspectives—A Study of an Intranet and an Organisation, Master's thesis, Royal Institute of Technology, Stockholm, Sweden (2005).

[78] H. Kasper, Redesign von Benutzungsoberflächen durch Mittel der Navigation [Redesigning User Interfaces by the Means of Navigation], Master's thesis, Universtiy of Stuttgart, Germany (2003).

[79] H. Kavakli, A Course Content Management System Development and its Usability, Master's thesis, Middle East Technical University, Ankara, Turkey (2004).

[80] M. Khan, A. Ahmad, Usability Evaluation of a Hypermedia System in Higher Education, Master's thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden (2008).

[81] D. Kieras, Model-Based Evaluation, chap. 58, in: Jacko and Sears [69], pp. 1139–1151.

[82] J. Kirakowski, N. Claridge, R. Whitehand, Human Centered Measures of Success in Web Site Design, in: Proceedings of the 4th Conference on Human Factors & the Web, Basking Ridge, NJ, USA, 1998.

[83] P. Koutsabasis, T. Spyrou, J. Darzentas, Evaluating Usability Evaluation Methods: Criteria, Method and a Case Study, in: Proceedings Part I of the 12th International Conference, HCI International, Beijing, China, 2007.

[84] S. Krug, Don't Make Me Think, Macmillan USA, 2000.

[85] M. Kuniavsky, Observing the User Experience, Morgan Kaufmann Publishers, 2003.

[86] L. Leventhal, J. Barnes, Usability Engineering, Pearson Prentice Hall, 2007.

[87] M. D. Levi, F. G. Conrad, A Heuristic Evaluation of a World Wide Web Prototype, Interactions 3 (1996) 50–61.

[88] J. Lewis, Sample Size for Usability Studies: Additional Considerations, Human Factors 36 (1994) 368–378.

[89] J. R. Lewis, IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use, Tech. Rep. 54.786, Human Factors Group, Boca Raton, FL (1993).

[90] R. Liskowsky, B. Velichkovsky, Wünschmann, Software Ergonomie '97 [Software Ergonomics '97], Teubner, 1997.

[91] M.-L. Liu, Photoware Interface Design for Better Photo Management, Master's thesis, School of Informatics, Indiana University, Indianapolis, IN, USA (2005).

[92] J. Looser, AR Magic Lenses: Addressing the Challenge of Focus and Context in Augmented Reality, Ph.D. thesis, University of Canterbury, New Zealand (2007).

[93] J. M. López, I. Fajardo, J. Abascal, Towards Remote Empirical Evaluation of Web Pages' Usability, in: Proceedings Part I of the 12th International Conference, HCI International, Beijing, China, 2007.

[94] P. J. Lynch, S. Horton, Web Style Guide: Basic Design Principles for Creating Web Sites, B & T, 1999, 2nd Edition Available Online, Retrieved July 15, 2008 from http://www.webstyleguide.com/index.html?/contents.html.

[95] R. Mahajan, A Usability Problem Diagnosis Tool—Development and Formative Evaluation, Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA (2003).

[96] K. Maier, Konzipierung und Implementierung einer Online Hilfe für ein virtuelles Konferenzsystem im Rahmen des von der Europäischen Komission geförderten Projektes "Invite EU" [Conception and Implementation of an Online Help System for a Virtual Conference System within the Project "Invite EU", funded by the European Commission], Master's thesis, College of Higher Education of Stuttgart, Germany (2000).

[97] M. Manhartsberger, S. Musil, Web Usability: Das Prinzip des Vertrauens [Web Usability: the Principle of Trust], Galileo Press, 2001.

[98] Microsoft Corporation, The Windows Interface Guidelines for Software Design: An Application Design Guide, Microsoft Press, 1995.

[99] R. Molich, J. Nielsen, Improving a Human-Computer Dialogue, Communications of the ACM 30 (3) (1990) 338–348.

[100] A. Moll, BALLView, a Molecular Viewer and Modelling Tool, Ph.D. thesis, University of the Saarland, Germany (2007).

[101] M. J. Muller, L. Matheson, C. Page, R. Gallup, Participatory Heuristic Evaluation, Interactions 5 (5) (1998) 13–18.

[102] M. J. Muller, A. McClard, B. Bell, S. Dooley, L. Meiskey, J. Meskill, R. Sparks, D. Tellam, Validating an Extension to Participatory Heuristic Evaluation: Quality of Work and Quality of Work Life, in: Proceedings of ACM CHI '95 Conference on Human Factors in Computing Systems, Denver, Colorado, USA, 1995, pp. 115–116.

[103] E. Möllmann, Computergestützte Informationssysteme im Museum [Computer-Based Information Systems in the Museum], Ph.D. thesis, University of Bielefeld, Germany (2007).

[104] NASA, NASA TLX: Task Load Index - Project Homepage, accessed July 15, 2008 at http://humansystems.arc.nasa.gov/groups/TLX/index.html.

[105] J. Nielsen, Website of Jakob Nielsen, accessed June 28, 2008: http://www.useit.com/.

[106] J. Nielsen, Usability Engineering, Academic Press, 1993.

[107] J. Nielsen, Enhancing the Explanatory Power of Usability Heuristics, in: B. Adelson, S. Dumais, J. Olson (eds.), Proceedings of ACM CHI '94 Conference on Human Factors in Computing Systems, 1994, pp. 152–158.

[108] J. Nielsen, Heuristic Evaluation, chap. 2, in: Nielsen and Mack [114], pp. 25–62.

[109] J. Nielsen, Usability Inspection Methods, in: CHI '95: Conference Companion on Human Factors in Computing Systems, 1995.

[110] J. Nielsen, The Use and Misuse of Focus Groups, Software, IEEE 14 (1) (1997) 94–95.

[111] J. Nielsen, Designing Web Usability: the Practice of Simplicity, New Riders, 2000.

[112] J. Nielsen, H. Loranger, Web Usability, Addison Wesley, 2006.

[113] J. Nielsen, R. L. Mack, Executive Summary, chap. 1, in: [114], pp. 1–23.

[114] J. Nielsen, R. L. Mack (eds.), Usability Inspection Methods, John Wiley & Sons, New York, 1994.

[115] J. Nielsen, R. Molich, Heuristic Evaluation of User Interfaces, in: J. C. Chew, J. Whiteside (eds.), CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems, 1990, pp. 249–256.

[116] J. Nielsen, M. Tahir, Homepage Usability - 50 enttarnte Websites [Homepage Usabiliy: 50 Websites Deconstructed], Markt & Technik Verlag, 2002.

[117] D. A. Norman, The Design of Everyday Things, The MIT Press, 1988.

[118] K. L. Norman, B. Shneiderman, QUIS Project Homepage, Accessed July 15, 2008 at http://lap.umd.edu/quis/.

[119] C. L. North, A User Interface for Coordinating Visualizations based on Relational Schemata: Snap-Together Visualization, Ph.D. thesis, University of Maryland, USA (2000).

[120] K. Oertel, Strategien zur Bewertung der Gebrauchstauglichkeit von interaktiven Web Interfaces [Strategies for Evaluating the Usability of Interactive Web Interfaces], Ph.D. thesis, University of Rostock, Germany (2003).

[121] R. Opperman, B. Murchner, H. Reiterer, M. Koch, Software-ergonomische Evaluation. Der Leitfaden EVADIS II [Evaluation in Terms of Software Ergonomics—The EVADIS II Compendium], Walter de Gruyter, 1992.

[122] R. Parizotto, Elaboração de um Guia de Estilos para Serviços de Informação em Ciência e Tecnologia Via Web [Development of a Web Styleguide for Information Services in Science and Technology], Ph.D. thesis, Universidade Federal de Santa Catarina, Florianópolis (1997).

[123] F. Paternò, Model-Based Design and Evaluation of Interactive Applications, Springer-Verlag, 2000.

[124] M. Pearrow, Web Site Usability Handbook, Charles River Media, 2000.

[125] D. Pierotti, Heuristic Evaluation - A System Checklist, Xerox Corporation, retrieved July 8, 2008 from http://www.stcsig.org/usability/topics/articles/he-checklist.html (1995).

[126] H. Pradeep, User Centered Information Design for Improved Software Usability, Artech House, 1998.

[127] J. Prümper, IsoNorm Questionnaire, accessed July 15, 2008 at http://www.ergo-online.de/.

[128] J. Prümper, Der Benutzungsfragebogen ISONORM 9241/10: Ergebnisse zur Reliabilität und Validität [The Questionnaire ISONORM 9241/10: Results of Reliability and Validity], in: Liskowsky et al. [90], pp. 253–262.

[129] V. Redder, Medienergonomische Gestaltung von online Informtaionssystemen des Typs "Register" [Media Ergonomic Design of Online Information Systems, Type "Register"], Ph.D. thesis, University of Bremen, Germany (2002).

[130] M. Reichel, Konzeption, Entwicklung und Usability Evaluation einer Webanwendung für die Verwaltung von Webhosting Leistungen [Conception, Development, and Usability Evaluation of a Web Application for the Administration of Webhosting Services], Master's thesis, Technical and Economical University of Dresden, Germany (2006).

[131] L. Rosenfeld, P. Morville, Information Architecture for the World Wide Web, O'Reilly Associates, 1998.

[132] J. Rubin, D. Chisnell, Handbook of Usability Testing, Wiley Publishing Inc., 2008.

[133] F. Sarodnick, H. Brau, Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung [Usability Evaluation Techniques—Scientific Fundamentals and Practical Applicability], Huber, Bern, 2006.

[134] J. Sauro, E. Kindlund, A Method to Standardize Usability Metrics into a Single Score, in: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems, 2005.

[135] I. Scandurra, Building Usability into Health Informatics—Development and Evaluation of Information Systems for Shared Homecare, Ph.D. thesis, Uppsala University, Sweden (2007).

[136] K. Schmid, Evaluation, Konzeption und Modellierung eines mobilen Informationssystems mit J2ME für den Einsatz bei Sportveranstaltungen am Bsp eines Golfturniers [Evaluation, Concept, and Model for a Mobile Information System with J2ME for the Use at Sporting Events Using Golfing Tournaments as an Example], Master's thesis, University of Koblenz-Landau, Germany (2006).

[137] A. Sears, Heuristic Walkthroughs: Finding the Problems Without the Noise, International Journal of Human-Computer Interaction 9 (1997) 213–234.

[138] H. Sharp, Y. Rogers, J. Preece, Interaction Design, John Wiley & Sons, Ltd, 2007.

[139] B. Shneiderman, C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley, 2004.

[140] A. Singh, A Guide to Improving the E-Commerce User Interface Design, Master's thesis, Durban Institute of Technology, South Africa (2005).

[141] S. L. Smith, J. N. Mosier, Guidelines for Designing User Interface Software, Tech. Rep. ESD-TR-86-278, MITRE Corporation (1986).

[142] J. M. Spool, Web Site Usability—A Designer's Guide, Morgan Kaufmann Publishers, Inc., 1999.

[143] D. Stone, C. Jarrett, M. Woodroffe, S. Minocha, User Interface Design and Evaluation, Morgan Kaufmann Publishers, 2005.

[144] S. Stowasser, Methodische Grundlagen der softwareergonomischen Evaluationsforschung [Methodical Fundamentals of Software-Ergonomic Evaluation Research], Shaker Verlag, 2006.

[145] Sun Microsystems Inc., Sun Web Styleguide (2000), http://www.sun.com/styleguide [offline since November 2000].

[146] Sun Microsystems Inc., Open Look: Graphical User Interface Application Style Guidelines, Addison-Wesley, 1990.

[147] Sun Microsystems Inc., Java Look and Feel Design Guidelines, Advanced Topics, Addison-Wesley Longman, 2001, available Online: Retrieved July 15, 2008 from http://java.sun.com/products/jlf/ed2/book/.

[148] D. Te'eni, J. Carey, P. Zhang, Human Computer Interaction, John Wiley & Sons, Inc, 2007.

[149] The Human Factors Research Group Ireland, HFRG Website, accessed July 18, 2008 at http://www.ucc.ie/hfrg/index.html.

[150] F. Thissen, Screen-Design Handbuch [Handbook on Screen Design], Springer-Verlag, 2001.

[151] B. Tognazzini, The First Principles of Interaction Design, retrieved June 25, 2008 from http://www.asktog.com/basics/firstPrinciples.html.

[152] T. Tullis, B. Albert, Measuring The User Experience, Morgan Kaufmann Publishers, 2008.

[153] US Department of Health and Human Services, Research-Based Web Design & Usability Guidelines (2006), retrieved July 20, 2008 from http://www.usability.gov/guidelines/.

[154] Usability Laboratory of the University Santa Catarina, ErgoList Project, accessed July 15, 2008 at www.labiutil.inf.ufsc.br/ergolist/.

[155] K. Vogler, Usability von Web Content Management Systemen - Analyse von Verbesserungspotentialen im Bereich der Usability [Usability of Content Management Systems—Analyzing Potential Usability Enhancements], Master's thesis, University of Applied Sciences Burgenland, Austria (2006).

[156] W3C, Web Content Accessibility Guidelines (1999), Online Version, accessed July 15, 2008 at http://www.w3.org/TR/WCAG10/.

[157] A. Westerberg, Evaluation of the User Interface of a Web Application Platform, Master's thesis, Umeå University, Sweden (2006).

[158] C. Wharton, J. Rieman, C. Lewis, P. Polson, The Cognitive Walkthrough Method: A Practitioner's Guide, chap. 5, in: Nielsen and Mack [114], pp. 105–140.

[159] C. Wiberg, A Measure of Fun—Extending the Scope of Web Usability, Ph.D. thesis, Umeå University, Sweden (2003).

[160] L. Wroblewski, E. M. Rantanen, Design Considerations for Web-Based Applications, in: Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society, 2001.

[161] K.-P. Yee, K. Swearingen, K. Li, M. Hearst, Faceted Metadata for Image Search and Browsing, in: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems, Amsterdam, The Netherlands, 2003.

[162] P. Zhang, G. M. von Dran, Satisfiers and Dissatisfiers: A Two-Factor Model for Website Design and Evaluation, Journal of the American Society for Information Science 51 (2000) 1253–1268.