Chapter 13
Methodology III: Empirical Evaluation

Abstract Evaluation is an essential part of development. There are several good reasons for carrying out user testing in particular. A successful evaluation requires careful planning. Here we describe the issues that you need to take into account and discuss several effective methods that can be used to collect data. User testing reduces the risk that you will deliver a system to your users that is unusable and is therefore ineffective. We also touch briefly on the need make sure that any evaluation that you carry out is conducted according to appropriate ethical guidelines.

13.1 Introduction

Evaluation should be a routine and regular part of system development. Attitudes to evaluation continue to vary widely, however, with many people still mistakenly believing that testing only happens at the end of development. The main problem with such an approach is that if development overruns while the delivery deadline remains fixed, it is usually testing that gets cut (because it is the last thing to happen). The net result is that the delivered system ends up being tested inadequately and, as a result, does not work as expected.

There are two statements that are worth remembering when it comes to evaluation. The first is "Test early, test often." The second is "Quick and dirty is better than nothing." Any evaluation program should deliver something informative for the system design and development. Testing can be expensive, but need not be. Lightweight methods that offer a proportionately larger return on investment are also available (Monk 1998; Nielsen 1993).

It is also worth remembering that when software engineers, in particular, talk about evaluation they will often refer to V & V, or verification and validation. Verification is about making sure that you are building the product right, and it is a process that is often carried out within the developer company. In contrast, validation is about making sure that you are building the right product, and usually

F. E. Ritter et al., *Foundations for Designing User-Centered Systems*,
DOI: 10.1007/978-1-4471-5134-0_13, © Springer-Verlag London 2014

involves some sort of acceptance test with the customer to make sure that the requirements have been met. You should make sure that you have the distinction between these two clear in your own head.

In this chapter we provide an introduction to the topic of evaluation and the methods that can be used to collect data. For a much fuller discussion about which analytical methods to use, you should consult an appropriate textbook such as Lazar et al.'s (2010) book *Research methods in human–computer interaction*. We do not go into the details of the methods you need to use to analyze your data. The way that you analyze the data is important, but is highly dependent on the type of data that you collect, and the overall purpose of the evaluation. More information about how to analyze your data can be found in statistics books, such as Howell (2010) or, for qualitative data Miles et al. (2013) and Todd (2004).

13.1.1 Why Do We Need User Testing?

If we could always guarantee that systems would be built correctly, based on a complete and accurate analysis of user requirements, then there would not (theoretically, at least) be a need for user testing. There are several reasons why we cannot give such guarantees, however. Perhaps the most pertinent one here is what has been called the *envisioned world problem* (Carroll and Rosson 1992; Woods and Dekker 2000), which is based on the fact that the world will inevitably change between the time when the user requirements are analyzed and the time when the final system is delivered. Often the changes are in ways that cannot be (fully) controlled by the users. So, when the requirements analysis is carried out, the users are effectively being asked to define what the system will need to do to work acceptably in some future context which they cannot fully predict or envision.

There are still some system developers who believe that as long as a system does what they consider to be the right thing, then that is enough. Their attitude is that if the users cannot make the system work, it is due to the users' lack of knowledge or ability, rather than being due to some fault of the developers. The fundamental problem with this view is that if the delivered system does not fit with the way that users normally carry out their work, there is a risk that it may not be accepted (e.g., see Berg 1997). The acid test of acceptability often comes when the users have to try to use the new system in their normal working environment.

The purpose of user testing is not to understand users but to evaluate how particular users can carry out particular tasks (using your system) in a particular context. There are still some designers who mistakenly believe that they know exactly how users carry out their work. In practice, however, when you observe users you see that they are often very creative, and use systems in ways that were never conceived of by their designers. The use of spreadsheets to format text or to make art, the use of thumbs to press cell phone buttons, and the use of social media broadcast tools like Twitter to organize political activities are just a few examples of creative use that were never imagined by their designers.

We know a lot about users, and their capabilities and limitations. We have covered much of this in the first part of this book and considered the implications for system design. Given what we know about the human visual system (Chap. 4), we can deduce that a 3-point font is likely to be too small to read. User testing is not necessary here; the decision can be derived from first principles. User testing may still be essential, however, for other system related questions—for example, to provide evidence that the right design choices and assumptions have been made when theories and assumptions make different suggestions, and that the system will work in its intended context.

Comprehensive user testing can be expensive and time consuming, but that does not mean that user testing should be ignored. As noted above, there are several discount or lightweight methods that can be used (Monk 1998; Nielsen 1993) to identify potential problems more cheaply, and at an earlier stage of development.

13.1.2 When Do We Carry Out User Testing?

Everybody makes mistakes—in the most general sense of the term—from time to time. This includes developers who may base their design decisions on insufficient or incorrect information at any stage of the development process—for example during analysis, design, and/or implementation. The consequences of these mistakes may not always be immediately apparent, but may surface much later. Carrying out evaluations helps to identify potential problems so that they can be appropriately rectified before the system gets delivered.

The ultimate goal of user testing is to make sure that the users can get the delivered system to do what they want it to do. In addition to helping to show that a system meets its requirements and is acceptable to the users, testing can also help to identify flaws in the development process. User testing is therefore often an effective way of providing designers with timely feedback at a relatively early stage during ongoing iterative development. Evaluating cheap, simple prototypes early in the development cycle can help to avoid potentially costly mistakes at a later stage in the lifecycle.

It is worth noting one of the practicalities of design that has implications for evaluation. It may not always be possible for designers to consult users during all stages of development. Instead, they work with a customer representative to come up with a system that will be acceptable to the users. Sometimes, however, the customer representative is not one of the target users, so they may not fully understand the subtleties of how the users really carry out their work using the system. It is therefore better to try to make sure that you can involve real users during evaluation.

Many developers still neither really understand what their users do, nor why they do what they do. Design and development are often carried out in isolation from the end users—agile methods can help address this limitation—with the result that the delivered system is not acceptable, because it does not fit in with the way that the users work. User testing during the early stages of development can highlight potential problems: making developers watch users wrestle with their product in an effort to make it do what they want is often a real eye-opener.

At least some part of systems development is based on the use of guidelines and principles. There are many guidelines and principles available, and they are mostly generic, which can make selecting the relevant ones for a specific project a major task. If you do use guidelines and principles, user testing can help to highlight the more subtle aspects of the system's context that need to be taken into account during development, thereby providing evidence to support your particular choice.

13.2 Planning Your Evaluation Study

Collecting data is not difficult. Collecting the *right* data—data that is pertinent to the questions that are being posed—is not so straightforward. Collecting the appropriate data to answer strategically relevant questions requires careful planning. Here we identify some of the main issues that can help to improve the chances of your evaluation study being a success. Many of the issues that we highlight are central to good experimental design.

13.2.1 What Type of Data: Qualitative or Quantitative?

One of the first things you will need to consider is the type of data that you will collect. The general rule of thumb is that you should think about collecting data to help you understand the thing (system, product, application, and so on) before you collect data to more precisely measure it. Typically the first type of data to collect is qualitative data, in other words, statements or general behavior, rather than precise numbers. Quantitative measures (e.g., times, number of clicks) can be used to verify assumptions with some degree of confidence over the intended population of users.

13.2.2 Selecting a Hypothesis

Creating hypotheses for your evaluation study helps to frame the study but also to keep it focused and grounded. A hypothesis is a proposition of what you believe to be the case (e.g., that a change you have made will cause a change in user behavior in some way) and will check with data. The null hypothesis (H_0) normally states that there is no difference between the things you are testing, e.g., there is no difference between the usability of application A and of application B. The alternative hypothesis (H_1) and the null hypothesis are mutually exclusive, so in our case it would be that there *is* a difference between the usability of application A and of application B. The multi-dimensional nature of usability means that experiments are often set up with several pairs of hypotheses, one for each dimension of usability. Some user studies might have multiple hypotheses to be tested.

13.2.3 Identifying the Dependent and Independent Variables

An independent variable is a factor that is independent of user behavior, and which can be varied by the person carrying out the evaluation (or the experiment, more generally). The dependent variable is the thing that depends on the user's behavior, or on the changes in the independent variable(s). Within user centered design, the sorts of things that would often be considered as independent variables include the type of input device (e.g., touch screen or keyboard), or different web page designs. The dependent variables are often more limited, however, to those things that are generally taken to measure the usability of a system (e.g., efficiency, effectiveness, satisfaction, ease of learning, and workload).

13.2.4 What Type of Evaluation: Formative or Summative?

There are two basic types of user-based evaluation: formative and summative. Each of these has a different purpose, and takes place at a different stage of development, as described below. They each help to remove different types of uncertainty, and to reduce the risk that the system will not be usable, or will be unacceptable to the end users.

Formative evaluation can take place at any point during development. It is used to help designers refine and form their designs. The focus of formative evaluation is to identify problems and potential solutions. In this type of evaluation the desired result is an indication of any problems that there may be in using the system, possibly with some indication of their frequency of occurrence. The designers can then use these frequencies to help rate the severity of the problems so a decision can be made about which problems should be fixed first.

Summative evaluation is concerned with assessing the success of the finished system or product, summarizing its overall impact and effectiveness. It is often used to test for any fixes that may be needed before the system is released, and to assess future releases. The end result may be some sort of usability score, and individual organizations may have their own particular threshold values for acceptability. One useful metric is to require that novice users of a new system are able to demonstrate performance levels that are some predetermined percentage of expert levels on the same system. The performance levels are measured using a set of predefined benchmark tasks. Defining a set of appropriate summative measures remains a difficult problem to solve, however, because systems, unlike many manufactured goods, cannot be assessed against tight specifications and tolerances.

13.2.5 Validity, Reliability, and Sensitivity

When it comes to designing an evaluation study, you want to make sure that you are evaluating the right thing, that you can measure the effects that you are looking for, and that the results can be generalized to other situations. To achieve these goals, you will need to think about the issues of validity, reliability, and sensitivity. This is true irrespective of whether you are collecting qualitative or quantitative data.

13.2.5.1 Validity

Validity refers to whether the measure that you are using is really measuring what it is supposed to be measuring. Reliability, on the other hand, refers to the consistency of a measure across different conditions. Note that it is possible to use a measure that has validity, but is not reliable, and vice versa. You should be aiming for high degrees of validity *and* reliability.

In addition to validity and reliability, you will also need to consider the sensitivity of the measure that you are using: does it react sufficiently well to changes to the independent variable. Validity, reliability, and sensitivity will all differ, depending on the context in which you are doing the evaluation.

There are several types of validity that you will need to think about. Here they are classified into two basic types:

- Instrument validity which relates to the instruments or measures that you will use in your evaluation. There are three subtypes: construct validity, content validity, and face validity.
- Experimental validity which relates the generalizability of the results. There are three subtypes: internal validity, external validity, and ecological validity.

We discuss each of these in more detail below as well as explaining the trade-offs that you may need to make when deciding on which type of experimental validity is important to your evaluation study.

Construct validity refers to the extent that your instrument or measure really does measure what you think it does. Probably the simplest example is to think about an IQ test as a whole, and how much it actually measures intelligence. Supporting evidence usually comes from both theory and testing, and can include statistical analysis of how responses and test items are related. If you think about usability, and how you measure that, things start to get a bit trickier because there are several different dimensions to the concept of usability. In other words, you cannot directly assess the usability of an artifact using a single measure. In this case you first have to operationalize the concept of usability, and then measure the different dimensions separately to assess efficiency, effectiveness, satisfaction, and so on. So although you are not measuring usability directly, by measuring the separate dimensions you are improving your construct validity. At this stage you will also need to think about the *content validity*.

Content validity refers to whether the content of a measure or instrument corresponds to the content of the construct that the test was designed to cover. Again, if we think about an IQ test, its content validity is determined by whether the items in the test cover all the different areas of intelligence that are discussed in the literature. For a usability survey, you would systematically examine the items in the survey to make sure that you had covered all of the relevant aspects of usability for the artifact that you are evaluating. So you might have items about the display layouts, the content, and so on. Often the way that content validity is evaluated is by having domain experts compare the test items against the specification for the thing that is being tested.

Face validity (also called surface validity) refers to whether a test *appears* to measure a certain criterion. It is closely related to content validity. The main difference is that you assess content validity by using a systematic review, whereas you assess face validity by having people make judgments about the test simply based on the surface appearance of the test. You could assess face validity, for example, by asking somebody (it does not have to be an expert) what they think the test is measuring. Sometimes you may get more honest answers if you have lower face validity, because the people doing the test are focused more on the task than what they think is being tested. It is also worth noting that you should not rely on face validity alone, because even so-called experts can get it wrong (consider, for example, the way they used to test whether someone was a witch or not in the Middle Ages).

Note that a test may have poor face validity but good construct validity. A game in which you shoot a gun at letters might not appear to measure spelling ability, for example, unless the letters pop up in a pattern that is based on correct spelling. Similarly, a tank simulation game might have poor surface validity for a naval task. However, both of these situations might have good construct validity in that the mental representations or the perceptual goals and cues are accurately represented (Smallman and St. John 2005). So you may have to consider how to trade off face validity against construct validity (and content validity) when designing your evaluation study.

Internal validity refers to how well conclusions can be drawn about cause-effect (causal) relationships based on the study design, including the measures used, and the situation in which the study was carried out. Internal validity is generally highest for tightly controlled studies which investigate the effect of an independent variable on a dependent variable, often run in a laboratory setting. To get good internal validity you need to make sure you control for other effects that could have an impact on the results you obtain. These include:

- Maturation of the participants: if their condition changes over the duration of the study, e.g., they become more tired.
- The types of participants: it is often impossible to use randomly selected participants, so you need to make sure that you do not end up with unbalanced groups of participants (e.g., all males, or all people aged over 50 in one of the groups).
- Testing effects: if you give participants in the study the same test at two different times, they may find it easier the second time because they already know the questions.

These potential confounding effects are well known. In many cases there are well documented solutions too which can be found in books on experimental psychology methods such as those by Calfe (1985), Ray (2008), and Campbell and Stanley (1963).

External validity relates to how far the results of the study can be generalized to other populations (such as different user groups), other places, and other times. One of the main factors that needs to be considered is the choice of participants used in the study. If all your participants are university students, for example, how can you be sure that the findings will apply to people who are aged over 60? To get good external validity you need to be aware of other effects that could have an impact on the results that you obtain, and provide some way of alleviating them. These effects include:

- Awareness of anticipated results: if participants can guess what they think the outcome of the study should be, they may adapt their behavior to what they think you expect them to do.
- The Hawthorne effect: people's performance can change simply as a function of being watched or recorded.
- Order effects: if you test people on artifact A first and then artifact B, there may be some carryover effect which means that their results with artifact B are better than they would otherwise have been.
- Treatment interaction effects: it may be the case that the participants in your study are motivated differently, and the way they are allocated to groups for testing means that you have one group that is more highly motivated than the other, which hence performs at a higher level.

Again, these potential confounding effects and solutions to the problems caused by them can be found in the literature mentioned above.

One way to increase external validity is to make sure that you use a wide range of users, stimuli, and contexts. The downside is that this will increase costs (time and money) and make it more difficult to detect real differences. Care is also needed to make sure that you do not reduce reliability (which is discussed below).

Ecological validity refers to the extent to which your results can be applied to real world settings. You should be able to see that it is closely related to external validity. For an evaluation study to have high ecological validity, the methods, materials, and setting of the study must approximate the real-life situation that is being investigated. If you wanted to conduct an evaluation of the usability of an application that is used in an office environment, for example, you would need to make sure that your setting resembled that of a normal office, where telephone calls and conversations (work and non-work related) are both constant sources of interruptions to task performance. The downside of having high ecological validity is that you cannot control all the possible independent variables that may affect the thing you are trying to measure.

At first glance there appears to be a conflict between internal and external (and ecological) validity. The main reason for carrying out evaluations in a laboratory setting is so that you can control for all interfering variables. Although this will increase your internal validity you lose external and ecological validity because you are using an artificial context for collecting data, and your results may not generalize to the real world—you may lose external and/or ecological validity. If you are carrying out an evaluation in a real world setting, however—using observation, for example—you will have high external (and ecological) validity but your internal validity will be reduced. Whether this is a problem or not depends on your research strategy. If you are following an inductive research strategy, then it is a problem because you will be concerned with the generalization of results; if you are following a deductive strategy, to test a theory, for example, then it is not a problem, because you are only concerned with threats to internal validity.

13.2.5.2 Reliability

Reliability refers to the ability of a measure to produce consistent results when the same things are measured under different conditions. Usually this is used in the context of *test–retest reliability*. In other words, if you conducted the same test again under the same conditions, but on a different day or with a similar set of participants, for example, you should get the same results if the measure is reliable. Reliability is also used in the context of assessing coding schemes, particularly when you need to encode the responses that you collect from users. If a coding scheme is reliable, then when you give the scheme and the data to another person, they should code the same data items in the same way. The level of agreement between the people who do the coding is what is called the *inter-rater reliability*, and you can measure this statistically (Cohen's Kappa test is often used to calculate the results).

13.2.5.3 Sensitivity

Even if the selected measure is both valid and reliable, it may not be sensitive enough to produce discernible effects that can easily be measured. The chosen measure may not change very much when you change the independent variables, for example. In this case it may be necessary to use a large number of participants. To achieve results that are statistically significant, however, you will still need to make sure that the measure has high reliability; otherwise your results will still be open to question.

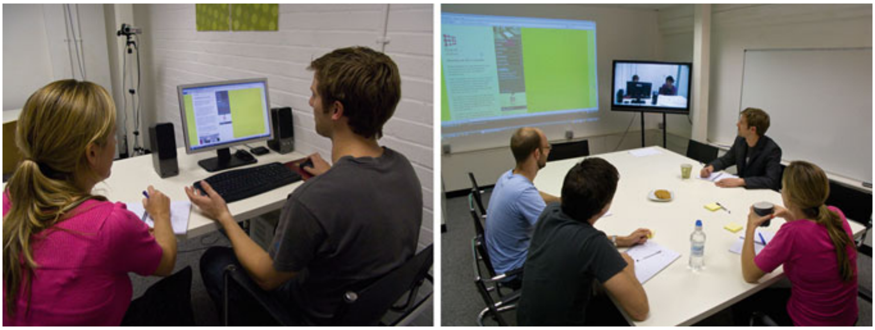

Fig. 13.1 A user working with a system being tested (*left*) and the observation room where designers and analysts can watch the study going on (*right*). (Copyright © Fluent Interaction Ltd, www.fluent-interaction.co.uk, reproduced with permission)

13.3 Evaluation Methods

Evaluation methods are generally divided into four categories: usability testing, field studies, expert (heuristic) evaluation, and A/B testing. One of the fundamental notions behind expert evaluation is that a small number of experts can be used to quickly identify a large number of the problems with the system. You should think carefully before using these methods because they are not ideally suited to all types of system. Each of the evaluation methods has its own strengths and weaknesses, and they are often used in combination with each other.

13.3.1 Usability Testing

The term usability testing is usually restricted to describing the evaluation of the usability of a system under controlled (laboratory) conditions. Quite often usability testing is carried out in dedicated laboratories that have been specially designed for the purpose. Fluent Studios, for example, is a usability laboratory based in central London. It consists of two purpose-built usability rooms, equipped with high definition audio-visual technology. These rooms are shown in Fig. 13.1.

Different organizations will set up their usability testing facility in different ways. Generally, one room is set up for testing whilst the other is configured as an observation room, which allows the developers, testers, and other interested stakeholders to see what the users are doing without disturbing them through their presence in the same room. The rooms are physically separated rather than being connected by a one-way mirror. Any action that takes place in the testing room is projected into the observation room. There are several advantages to using projection rather than a one-way mirror:

- Observers feel less intrusive
- Observers can talk in the observation room
- Observers can move around, allowing brainstorming activities
- Participants are less aware of being watched (because there is no large mirror in the room, and sound contamination is reduced)
- A visual area larger than the participant's screen can be used to observe what they are doing
- There is greater flexibility in the way the observation room can be configured
- Costs are reduced
- It is easier to use
- The volume of what the participant is saying can be controlled within the observation room.

Fluent Studios' laboratory is used to conduct usability tests in a creative environment. Whilst one-to-one testing is carried out with users in the testing room, it is regarded as very important to get the observers involved as well. So, often when a new design or prototype is being tested, the tests will be designed to try and identify usability issues early on in the testing, so that they can be resolved at the earliest opportunity.

Usually the first step in testing is to develop a set of task scenarios that capture the critical characteristics of the tasks that are likely to be carried out using the system (Carroll 2000). These scenarios are usually descriptions of real-world tasks that users can be expected to understand, but the scenario does not describe *how* the task is done using this system. They are typically expressed as problems that the user would normally be expected to solve using the system as part of their work.

For example, consider the following scenario:

> You are a system administrator for a software system that schedules and allocates resources ranging from company pool cars to meeting rooms. Unfortunately one of the meeting rooms has unexpectedly been designated to be refurbished, which will take 2 months beginning in July. Your task is to notify those people who have booked the room for July and August and to provide alternative resources.

You should be able to see that this scenario contains a goal, information about that goal, and information about the context in which the task takes place. It does not, however, contain instructions about how to use the system to achieve the desired goal.

The focus of the scenarios determines the shape of the evaluation: everyday usage scenarios, for example, will capture information about everyday usage of the system. Similarly, for critical systems (safety critical, mission critical, business critical, and so on) the scenarios would be designed to focus on critical (but unusual) incidents. Ideally a more comprehensive evaluation could be carried out using both types of scenarios.

An illustration of the benefits of usability testing occurred when new designs for a national educational site were being tested in Fluent Studios. Some usability problems were quickly observed with the method of global navigation: the first four users who were tested all struggled to find the site's home page. Between testing sessions a new page header was developed, and tests with the next four participants demonstrated that the introduction of the new design had resolved the problem. This rapid development methodology is used to test designs in an agile way, which makes the most effective use of the testing time allocated to a project.

13.3.2 Field Studies and Field Experiments

Field studies, as the name implies, are evaluations that are carried out in the field, that is, in real world settings. Field studies are often carried out to discover more about the context of use of a technology that is to be designed or is being designed. Studying activities in the "real" world can be challenging. If you just think about the things that happen every day in an office environment you should start to get the picture. In an office, users may be carrying out some task using the computer. They can break off from their work at any point, though, such as when they are interrupted by the telephone ringing, if a colleague stops by to discuss something (work or otherwise), or if their boss calls them into a meeting. So carrying out the task is not simply a matter of planning what to do, then just getting on and doing it step by step in sequence, from start to finish: people will often have to juggle several, possibly unrelated, and often unscheduled, activities at once. The big advantage of field studies is that they show you how people really work. One obvious design implication of the fractured nature of work is that you should make it relatively easy for people to pick up where they left off after an interruption. The main disadvantage of field studies is that it is often very difficult to exercise experimental control over what happens, so it is harder to focus on the relationships between some of the task variables, and to have results that are both general and applicable to other settings or times.

Field experiments are trials of technologies in real world settings. This is often when a fully functional prototype can be deployed into a real world setting and, as designers and developers, we want to see how users will interact with the technology. Such field experiments tend to be for non-safety–critical systems, such as recreational and social Internet sites. Often there is some latitude for changing the technology, but most of the functionality is set. In this instance, the evaluation will likely involve many different methods: collecting usage data, conducting observations of the technology in use, interviewing and surveying users, small controlled experiments, and so on (for an example of a technology that was fielded and evaluated see Churchill et al. 2003).

13.3.3 (Expert) Heuristic Evaluation

Heuristic evaluation (Nielsen and Molich 1990) is a relatively informal way of analyzing the usability of an interface design. A small select number of people—ideally interface design experts, and preferably domain experts too—are asked to

Table 13.1 Heuristic basis for user interface evaluation (adapted from http://www.nngroup.com/articles/ten-usability-heuristics)

1. The current system status should always be readily visible to the user
2. There should be a match between the system and the user's world: the system should speak the user's language
3. The user should have the control and freedom to undo and redo functions that they mistakenly perform
4. The interface should exhibit consistency and standards so that the same terms always mean the same thing
5. Errors should be prevented where possible
6. Use recognition rather than recall in order to minimize the mental workload of the users
7. The system should have flexibility and efficiency of use across a range of users, e.g., through keyboard short-cuts for advanced users
8. The system should be esthetic and follow a minimalist design, i.e., do not clutter up the interface with irrelevant information
9. Users should be helped to manage errors: not all errors can be prevented so make it easier for the users to recognize, diagnose, and recover
10. Help and documentation should be readily available and structured for ease of use

make judgments, based on a set of guidelines or principles together with their own knowledge, about a particular design. The individual results are then aggregated together. In this way it is possible to overcome the inherent inaccuracy of individual evaluations.

In the ideal world all of the evaluators would use the same (standard) set of criteria for judging what is good or bad. In reality, most people tend to rely on intuition and common sense, partly because most usability guidelines tend to be excessively large, often having many tens or even hundreds of rules (e.g., Brown 1988; Mosier and Smith 1986). Molich and Nielsen (1990), however, suggested that a relatively simple set of guidelines can be used as the basis for evaluation. Initially they used nine guidelines, but over the years, these have been refined and the number increased to ten, as shown in Table 13.1.

Each expert works their way through the user interface design individually noting compliance with the heuristics. Note that the user interface may just be a paper-based prototype, because the experts are not being asked to carry out tasks using the system. Problems that are detected can either be written down or recorded verbally (e.g., by taking verbal protocols). The individual results are then aggregated to highlight the detected problems.

Typically, only three to five experts are required to carry out a heuristic evaluation and generate useful results. Many people have taken this to be a hard and fast rule for all types of evaluation, however, which can be a serious mistake. The requisite number, to a large extent, depends on the diversity of the eventual user population. So, for example, if you were designing an on-line population census, then it would not make sense to just use three to five users, since such a small sample is very unlikely to be truly representative of the diversity inherent in a nation's general population.

It should be noted that many of the problems that are identified by heuristic evaluation may not affect the usability of the system. In addition, the results are often only presented in negative terms, focusing on what is bad about the design, instead of also highlighting the things that are good.

13.3.4 Co-operative Evaluation

Co-operative evaluation is another type of expert evaluation method. It was developed at the University of York (UK), and is related to Scandinavian design practices (Monk et al. 1993; Müller et al. 1997). As the name suggests, the evaluation is carried out co-operatively, with the user effectively becoming part of the evaluation team. The method is based on the notion that any user difficulties can be highlighted by two simple tactics:

1. Identifying the use of inefficient strategies by the user (e.g., copy-paste-delete, rather than cut and paste).
2. Identifying occasions when the user talks about the interface, rather than their tasks. These are called breakdowns, based on the notion that good tools should be transparent, so the user should be talking about the task rather than the technology.

The user is asked to talk aloud as they carry out a series of tasks, and can be prompted with questions. It is a formative evaluation technique, in that it is used to gather information about the design as it is being formed. The method can therefore be used with a working prototype or with the real system.

13.3.5 A/B Testing

A recent trend is to do live testing of multiple interfaces. This is called A/B testing or bucket testing. In this approach a web service exposes different users to different interfaces and/or interactions. This can be seen at Google, for example, who was one of the first Internet sites to use this method extensively to guide their interface and interaction design decisions. In bucket tests, interfaces can vary in subtle ways, such as color changes, or may differ substantially, including manipulations of key functionality. User actions such as clicks (measured as CTR or click through rates) are studied to see the impact on user behavior, if any, of the changes.

There are many advantages to these kinds of studies—not least that a test can be run at scale and while maintaining ongoing business, and that feedback is fast. Of course, this approach requires building the test interfaces, and having the platform on which to partition users into conditions and deliver the experiences.

13.4 What to Evaluate?

In general, the sooner evaluation is done during development, the sooner you will get feedback, and the more likely it is that the delivered product will be both usable and acceptable to users. There is a trade-off here between the need for evaluation and how closely related the current version is to the final version. If an iterative development life cycle approach is used, this means that evaluation should be carried out as part of each iteration. Obviously, the earlier in development that evaluation takes place, the less developed the system that is being evaluated will be. The way that evaluation is often carried out during development is using some sort of prototype, until eventually the full system can be evaluated. The prototype usually starts out as something with low fidelity (possibly just one or more sketches), and increases in fidelity as the project progresses.

13.4.1 Pencil and Paper Prototypes

At the earliest stages of development, pencil and paper mockups of the interface can be shown to the users, who are then asked how they would carry out particular tasks. This technique, which is often described as storyboarding should ideally be carried out with real users, although other designers, and even hostile users, could be employed.

Using pencil and paper sketches is cheap, and can be used very early in design. At this stage, ideas are usually still being explored, so the evaluations are usually formative. The data collected are normally qualitative rather than quantitative, with the results being used to inform the design of the artifact.

13.4.2 Computer-Based Prototypes

The next level of sophistication involves building a computer-based prototype. There are now several tools around that can be used for prototyping at several levels, from pencil and paper style sketches to interactive working prototypes.

At the earliest stages of computer-based prototyping you can employ *Wizard of Oz* techniques, where the user will interact with a prototype (computer-based) interface. The full functionality of the rest of the system is usually not available at this point, so a human acts behind the scenes to process user input and actions and provide the required responses (like the Wizard of Oz did in the film!). This technique has been used very effectively for evaluating the potential of speech-based systems.

The level of sophistication of the prototype should naturally increase as development progresses, and the prototype becomes closer to the finished product. How it develops will depend mostly on which basic method of prototyping you use: evolutionary or revolutionary. In evolutionary prototyping, the original prototype is refined after each iteration of the development cycle: the prototype evolves towards the deliverable version. This is the sort of approach that is used in developing web sites, where they use wireframes to lay out the basic initial design, which then gets filled in and refined as the design evolves. In revolutionary prototyping, the current prototype is thrown away at the end of each iteration of the development cycle, and a new prototype is developed.

In addition to helping identify design issues, prototypes can also be used to help users to articulate requirements. People often find it much easier to talk about something concrete, referring to the prototype, than to talk about something abstract, where they have to imagine what the application or product should do.

Prototypes vary in cost, depending upon the sophistication of the prototype and the length of the evaluation period (laboratory-based user testing vs field studies). They do tend to give good results and are suitable for many stages of the design process, for both formative and summative evaluations.

13.4.3 The Final System

Evaluations of the final system will often be performed in house first, possibly using laboratory-based testing. If there is latitude for some redesign, systems may be deployed and field experiments conducted, but this only tends to be the case for systems that are not safety–critical, as noted above. For enterprise, and safety- and security-critical systems, it is more usually the case that the system is evaluated in full before it gets delivered to the customer. The final system will usually be subjected to a formal acceptance test, normally on the customer's premises, where the customer will sign to say that the system has successfully passed the agreed tests. You should note that web sites are very rarely tested at the customer's site (largely because they will normally be used from elsewhere).

Once a system is delivered and has been accepted by the customer, it is unlikely that any further formal evaluation will take place. The picture is slightly different for web sites, however, where the delivered system will normally reside in a dynamic environment, so the iterative development may continue, albeit with iterations that have a longer duration. In both cases, data on patterns of usage may be collected, along with information about problems logged with customer support, as noted above, and this can be used to inform future development projects and refinements to existing systems and products.

13.5 Measuring Usability

There are several dimensions to usability, so there are several measures, both qualitative and quantitative, that can be used to indicate how usable a particular artifact is. Most people will be familiar with task time as the de facto standard for measuring efficiency and productivity, and hence giving an indication of usability. There are several others too, though, and they are usually concerned with either performance (quantitative measures) or process (qualitative measures). There is also the concept of user experience, related to how much satisfaction the user obtains from the system. Here we briefly describe the measures that you are most likely to encounter. Often you will use several complementary methods to measure usability, such as a combination of task performance times, and a usability survey.

13.5.1 Task Time

Task performance time is widely used as a measure of efficiency within the fields of HCI and human factors and ergonomics. The task is usually one of three: a small cognitively manageable task, often referred to as a *unit task* (Card et al. 1983); a standard, predefined benchmark task (that you use to assess efficiency for similar artifacts); or a task scenario (as described above).

It is easy to determine task time using a stop watch, for example, or using time stamps if you are recording task performance. Time is a measure that is widely understood, and is easy to analyze statistically. Time is generally used as a measure in summative evaluations of the final system. Where the performance time is relatively insensitive, however, it can be costly to carry out evaluations, because you will have to run the test many times to be able to draw solid statistical conclusions from the results.

Remember that usability is not only concerned with how easy something is to use, but also how easy it is to learn to use. Task times can also be used to determine how long it takes to learn to use a system. Normally some threshold level of performance is defined in advance, and the length of time it takes to reach that threshold is measured. Alternatively, the length of time it takes to recover from observable errors can be measured: you would expect to see this time reduce as people learn how to do the task and how to manage the errors.

One of the main problems of using time measures is that they are not easily compared unless all the contextual elements (tasks, level of expertise, lighting conditions, and so on) are kept constant. The corollary of this is that if you want to compare times when you cannot fully control the contextual effects, you have to convert the data into a more stable metric, i.e., one that is not so easily affected by changes in these elements. One way of doing this, which was proposed by Whiteside et al. (1985), is to calculate a score in the range 1–100 as shown in Eq. (13.1):

$$\mathrm{Score} = (1/\mathrm{T}) \times \mathrm{P} \times \mathrm{C} \tag{13.1}$$

where T = time, C = constant based on fastest expert time, and P = percentage of task completed.

13.5.2 Errors

Errors can be measured quantitatively (by simply counting them) or qualitatively (by noting the different types of error). Whilst time is best suited to summative evaluations, error measures can be used in both summative and formative evaluations.

As we have already seen in Chap. 10, however, errors are not easy to define, and they can be hard to count too. This is particularly true when observing expert behavior. One of the key aspects of expert performance is that they often detect and recover their own errors before the effects of the error become apparent to outside observers. So if you watch an expert perform a task, you may not even realize that they have made an error.

We can distinguish many types of errors—slips, mistakes, violations, mode errors (e.g., problems with grayed out menu items), discrimination errors (e.g., selecting the wrong menu item because of ambiguous labels), and so on. The types of errors will vary depending on which taxonomy of errors you use (see Chap. 10 for examples).

13.5.3 Verbal Protocols

Verbal protocols can be a useful way of understanding the issues that confront users as they try to tackle particular problems using some artifact. Some care is needed when reading about verbal protocols, because many people use the terms *talk aloud* and *think aloud* interchangeably. Strictly speaking you usually want people to produce *talk aloud* reports, reflecting the things that are in their short term memory as they do the task; if they generate *think aloud* reports, this suggests that they are processing things more deeply and (possibly) rationalizing their decisions and actions before they verbalize them.

The two main types of verbal protocols are concurrent, which are taken whilst the person performs the task, and retrospective, where the person describes what they did after completing the task. In concurrent verbal protocols, the user is asked to talk about information as it comes to mind, to "say out loud everything that you say to yourself" (Ericsson and Simon 1980, 1993). The user should not be reflecting upon their own behavior and providing explanations of causality. While this kind of reflective behavior (referred to as introspection) may provide some useful insights, these insights are not considered valid data because they are easily influenced by other aspects, such as expected task performance, social pressure, and the user's (often incorrect) theories of how their own mind works. Talking aloud about content is generally regarded as being more objective than thinking aloud, which usually involves introspecting about the process.

Providing concurrent protocols can be hard for users, but they are more reliable than other types of verbal protocol. When you take concurrent verbal protocols, you should ask the user to practice providing a concurrent verbal protocol whilst carrying out a simple task, such as an arithmetic addition or counting the windows in their childhood home (Ericsson and Simon 1993, appendix).

You may find it easier to collect concurrent protocols by having two users work together on a task. The natural dialogue that takes place (assuming that dialogue occurs or is required for the task) will encapsulate the information they are using to do the task. Another possible variation is to use expert commentary. Here one expert describes what the user is doing as they perform the task.

Retrospective protocols can also be used, and these are taken after the task has been performed. They tend to be more useful when people can watch a video or pictorial record—we discuss visual protocols in the next section—of their performance to help them remember what they did. This helps them to recognize their actions, rather than just having to recall them from memory. Although subjects may find it easier to provide retrospective verbal protocols, they can lead people to provide post hoc rationalizations of actions that they now perceive to be incorrect or that they performed instinctively.

Another way that you can interrogate what users are doing is by using pop-up menus (Feurzeig and Ritter 1988). This idea has not been fully tested, however, and does not have the same level of theoretical support as concurrent verbal protocols. The obvious criticism is that the pop-up menu interrupts the task, and may break the user's flow of activity because it draws their attention away from the task. A similar but more intrusive approach is to freeze the task and ask users about what they are doing at that particular point in time. This latter technique has been used in measuring situation awareness (Endsley 1995).

13.5.4 Video Protocols

Video protocols (also called visual protocols) involve making a video recording of users as they carry out some prescribed task. The recording is often made using multiple cameras positioned to capture different aspects of performance, such as what is currently shown on the screen, the position of the user's hands, and a more general view that shows both the user and the system together. Sometimes the recordings are made directly from the monitor. Although video protocols provide very rich data, the fact that they are being video recorded does make some users feel under pressure, and can lead to unnatural behavior.

The main problem with video protocols is that analyzing them can be very hard and is very time-consuming. Typically, analysis can take anywhere between 10 and 100 times as long as the duration of the recording.

As noted above, video protocols can be shown to users to help in the collection of retrospective verbal protocols. This technique is sometimes called auto-confrontation, because the users are shown the video recording and asked to explain their behavior.

Video protocols can be shown to developers to let them see the sorts of problems that real users encounter with their system. It is arguably better, though, to let the developers watch users try to use their product in a usability laboratory in real time. Both of these methods are generally much more effective than simply providing the developers with a written report of qualitative and quantitative performance data. They can also be used when the developers are remote from the site where the evaluation is taking place, as long as suitable network connections are available to transmit the recordings.

13.5.5 Eye Movement Tracking

In the last 10 years an increasing number of people have begun to collect data on eye movements to analyze how people use web pages (e.g., Nielsen and Pernice 2010; and see Navalpakkam and Churchill in press, for a more general review of eye-tracking). The current eye-tracking equipment is much easier to use, much cheaper, and much less invasive than earlier generations of eye-trackers which required you to have your head clamped in place, and required frequent re-calibration. They also generated large amounts of data that required significant effort to analyze and interpret, whereas there are now several good software packages available that will help you make sense of the data. You should recall that we discussed eye-tracking in Chap. 4.

Eye movement data is particularly useful as a way of generating heat maps which show the hot spots on a web page. These are the parts of a web page that users spend most of their time looking at, either by gazing at it for a long period of time, or visiting it for several shorter periods of time. In general, users have predetermined expectations about where they expect certain items such as menus, navigation bars, back/next buttons, and so on to appear on a web page. This leads them to automatically look for those items in the expected places first. If they are not where they are expected to be, you start to see scan patterns in the eye movements as the eyes jump around trying to find the required element.

There are some drawbacks to using eye movement data, which mean that you often need to complement it by using an additional method. The two main drawbacks are that the data do not tell you *why* users fixated on a particular point on the page and that the data do not tell you *what* items on the page the participant missed or did not notice.

13.5.6 Questionnaires and Surveys

If you want to discover opinions about something, often the best way is to ask people. Subjective measures are frequently used to assess attitudes towards a new piece of technology—feelings of control, frustration, etc. Sometimes just asking people for their opinions is the only way of gathering this data. Note, however, that sometimes surveys can measure opinions but not actions; early work has shown that what people do and what they say they will do can vary up to 100% (LaPiere 1934). Surveys are more valid when the attitudes are more stable, relevant, and salient to the behavior, and there are less situational pressures on the behavior (Hock 2002, pp 281–288). Questionnaires and surveys allow you to gather large amounts of data in a relatively short period of time, as long as you distribute them appropriately.

Designing questionnaires and surveys is an art in itself, as great care needs to be exercised to make sure that any potential biases are avoided. It is also important to make sure that the questionnaires are well structured and tested, as this helps to ensure the validity of the resulting data. For this reason, it is almost invariably a good idea to carry out a pilot study on a small sample of users, and then refine the questionnaires appropriately. Having a pilot study is also very useful for determining how long it will take to complete the survey. As a rule of thumb, most people are relatively happy with filling in surveys that take 10–15 min to complete, without any reward.

The questions need to be carefully designed, because you will not have a chance to explain them to respondents. So they need to be clear, unambiguous, and easy to understand. It is also important that you do not ask leading questions that reflect any biases that you may have. You also need to think about the answers that you require. In some cases it may be a simple "Yes/No/Don't Know," or it may be "select one (or more) options" from a possible list. In other cases (and quite often in usability surveys) you will be trying to gauge people's opinions about something, in which case you are more likely to use rating scales, such as a five-point Likert scale, where you will ask respondents how much they agree with a particular statement, such as "I found it easy to locate the home page button." In this case the response scale would normally be from *Strongly Disagree* to *Strongly Agree*.

Distribution of questionnaires and surveys requires careful thought. Usability surveys are frequently handed out on paper to participants as part of a usability study (often at the end). There are some standard usability rating scales that you could use or adapt for your own purposes, such as the System Usability Scale (SUS, Brooke 1996). More generally, however, you may want to use electronic surveys, in which case you need to think about how you will attract people from your target audience to complete the survey.

Note that if you intend to use follow-up surveys at the end of a test, you need to be aware of what is called *the media equation* (Reeves and Nass 1996). This refers to the fact that if you give people the survey on the same machine as the one on which you give them the test, they rate things more highly than if they complete the survey on a different machine! They treat the machine they used as an agent that needs to be treated socially.

13.5.7 Interviews and Focus Groups

Interviews can take three different forms: structured, unstructured, and semi-structured. Whichever type you decide to use, it is often a good idea to record them, with the written consent of the interviewees. It is also a good idea to make some written notes. These will help add extra context to help interpret the content that has been recorded, and will also act as a back-up in case recording fails for some reason.

Structured interviews are based around a fixed set of questions that the interviewees must answer. These questions are often closed, i.e., the user is expected to answer the question and no more. Typically these questions have “Yes/No” type answers.

Unstructured interviews are generally more informal, and are a bit more like a chat with the users. So you may start off with a small number of issues (perhaps as few as one or two) that you want to discuss with the users, and then the direction you take for the rest of the interview is determined by what they say.

Semi-structured interviews fall somewhere between structured and unstructured interviews. Usually you will have a short standard list of questions, which may be open, and then you direct the interview based on what the users say in response to the questions you ask. Unstructured and semi-structured interviews tend to be slightly harder to carry out because they will often require the interviewer to think on their feet during the interview. Their big advantage, however, is that they can uncover issues that may not previously have been thought of.

Whilst interviews tend to be carried out on a one-to-one basis, it can be useful to have group discussions, which are often carried out as focus groups. Usually a focus group is carried out with a small group of up to about ten users or stakeholders. The basic aim is to get the focus group members to express their opinions in a relatively friendly environment. To conduct a focus group successfully you need to have a list of issues or questions for discussion, and to have an experienced facilitator who can make sure that everybody gets a chance to air their opinions. The sessions can produce lots of useful data, so it is often best to record them as well as making notes (it may help to have separate people taking notes and facilitating the discussions).

13.5.8 Workload Measures

Workload measures attempt to describe how much mental effort the user expends in performing a particular task. They are generally used more often to evaluate critical systems rather than web sites per se. The measures are hard to devise, but can be useful in many contexts. The most common approach is to periodically ask users to state (or rate) what they think their current workload is, although this can be quite disruptive of performance and hence affect their perceived workload.

The NASA-TLX (Task Load indeX) workload measurement instrument (Hart and Staveland 1988) is probably the most commonly used method. NASA-TLX can be administered on paper or on-line. The NASA TLX is a multi-dimensional rating procedure that provides an overall workload score based on a weighted average of ratings on six workload dimensions: mental demands, physical demands, temporal demands, own performance, effort, and frustration (NASA, 1987).

During the standard NASA-TLX procedure users carry out pairwise comparisons of the six dimensions. In each of the 15 (5 + 4 + 3 + 2 + 1) comparisons, users select the dimension that contributed more to workload. Each dimension receives one point for each comparison where it was greater. The relative weight for each dimension is then given by the sum of those points, divided by 15 to normalize it.

Probably the most accurate approach for measuring workload is to use a secondary task that the user must perform as and when they can (e.g., responding to visual or auditory signals). For example, at random intervals the user has to push an ‘A’ when the number that pops up on the screen is odd and a ‘B’ when the number is even. The time and correctness of the response is a measure of how hard the user is working.

We sometimes find that while two systems give comparable performance results on the primary task, performance on the secondary task may be very different., This suggests that one interface is more demanding than the other, i.e., where performance on the secondary task is worse, this indicates that the user is expending more mental effort on the primary task.

13.5.9 Patterns of Usage

Rather than looking at performance on unit or benchmark tasks in a laboratory setting, you can place prototype versions of your system in real work settings and observe actual patterns of use, either directly or through videotape. Often you will find that certain features, including those that have been requested by users, are very rarely used, e.g., style sheets in Word.

You could also consider instrumenting the user interface or using a general keystroke logger (e.g., Kukreja et al. 2006) to collect (timed) logs of the keystrokes, and other interactions that the user performs. This data gets logged in what are sometimes called dribble files. These files can quickly become excessively large, however, and thus be hard to analyze. They can be used as a way to identify errors, error recovery, and patterns of use. Note that if you will be collecting data in this way, you will need ethical approval, which we talk about below.

If you are evaluating a system that has been released into the marketplace, you can also get some information on patterns of usage by looking at the logs of calls to customer/technical support services. Note that this data only measures problems that have been reported, rather than all of the problems. Users are often very flexible and adaptable and will develop ways of making the system do what they want it to do, such as workarounds, rather than spend extra time and effort on the end of a phone line trying to contact technical support to report the problem.

Customer support activity data can be both politically and commercially sensitive—it may allow competitors to see where the problems are with a particular product. Such data can be very valuable, however, because it does give a good indication of where the real problems may lie.

13.5.10 User Experience

Finally, there is the concept of user experience (Tullis and Albert 2008). In addition to these mostly quantitative measures, there is the qualitative experience of using the system that is being tested. This is an important concept, and is related to several concepts that can sometimes be hard to define, and even harder to measure. Many organizations now rate a high level of user experience (explained in Chap. 2) as being a major determinant in the success of a system or product.

One of the factors that will influence user experience is task importance. If the task is important to the user and the system gets the task done, then, it will be a successful system. Early systems, e.g., TVs, phones, portable phones, PDAs, Blackberries, were all hard to use and the times taken to use them were relatively high compared to today's standards. However, they were successful because they provided a better experience or supported a task that was not supported before. Over time and extended use, other measures and aspects became important.

13.6 The Ethics of Evaluation

Studies that involve users interacting with technological products now routinely need to be vetted to ensure that participants are treated appropriately. This means that ethical clearance (or approval) is required from the appropriate authoritative body. In most countries this will normally be done by an ethics committee, whilst in the US it will be carried out by an institutional review board (IRB). They will review the study to determine that the relevant guidelines are being followed. The main things they check are whether vulnerable people will be involved, whether participants are aware of what they are committing to in the study, and that any collected data is stored appropriately. Usually the latter involves anonymizing data so that they cannot be linked to the participant. These requirements vary based on funding, use, publication, and teaching, so take advice if you have not done this before.

As a matter of routine you should produce information sheets for participants, describing the study and explaining that they can withdraw from the study at any point. You should also take informed written consent, having them sign a consent form that says that they have read and understood the information sheet, that they are willing to take part, and that they understand that they can withdraw at any point. You should also think about how you will debrief at the end of a testing session: you could either give them a debrief sheet, explaining the purpose of the study in more detail, or simply verbally debrief them. You may also want to ask them not to discuss the study with others, because it could influence their behavior if they were subsequently in the study. Further details on this process are available (Ritter et al. 2013; Ritter et al. 2009).

13.7 Summary

There is a lot more to evaluation than many people imagine. Carrying out an evaluation requires careful thought and planning before you begin testing. In this chapter we have highlighted the sorts of issues you need to think about during planning. Most development is carried out using an iterative cycle in which a formative evaluation is carried out during each cycle. The information that comes out of the evaluation can then be used to inform development during the next cycle of development. It is therefore important that you clearly understand what sort of data you should collect, and why. You also need to think about whom you will collect the data from, and the environment in which you will collect them.

Once you have the basic plan, you can start to think in more detail about how you will collect the data (and how you will analyze them, although we have not covered that issue here). There are many methods that you could use, and your final choice may be determined by factors such as how much time is available, what resources are available, and how many (potential) users are accessible to take part in the tests. We have briefly discussed several evaluation methods that are available, and touched)on the importance of making sure that you are aware of the need to treat the participants in your evaluation in a way that is ethical.

Evaluation is important because it produces feedback on development as it is progressing. If you can get real users to take part in your evaluations, their feedback will help make sure that the system is more likely to be usable and acceptable when it is delivered. In other words, it will reduce the risks that the final system will be a failure when it is delivered.

13.8 Other Resources

There is a web site devoted to the subject of evaluating adaptive systems: EASy-Hub (which stands for Evaluation of Adaptive Systems Hub) is available at http://www.easy-hub.org.

If you are designing a usability laboratory, Jacob Nielsen edited a special issue of the *BIT* journal about how to create and use usability laboratories. Although it is now somewhat dated, it still contains several useful nuggets of information:

> Nielsen, J. (Ed.). (1994). Special issue: Usability Laboratories. *Behaviour & Information Technology 13*(1–2).

If you are not experienced working with studies with human participants, useful guides include:

> Ritter, F. E., Kim, J. W., Morgan, J. H., & Carlson, R. A. (2013). *Running behavioral studies with human participants: A practical guide*. Thousand Oaks, CA: Sage.

> Shadbolt, N. R., & Burton, A. M. (1995). Knowledge elicitation: A systematic approach.

In J. R. Wilson & E. N. Corlett (Eds.), *Evaluation of human work: A practical ergonomics methodology* (pp. 406–440). London: Taylor and Francis.

Stanton, N.A. & Young, M. (1999). *A guide to methodology in ergonomics: Designing for human use*. London, UK: Taylor & Francis.

One of the best introductions to user focused evaluations is by Elizabeth Goodman, Mike Kuniavsky, and Andrea Moed (also recommended in Chap. 2). They cover basic techniques and methods that will help you design better interactions. They also offer case studies and examples that you can compare to your own design situations:

Goodman, E., Kuniavsky, M., & Moed, A. (2012). *Observing the user experience: A practitioner's guide to user research.* San Francisco, CA: Morgan Kaufman

Another highly recommended text is Kim Goodwin's *Designing for the Digital Age*, published in 2011:

Goodwin, Kim. (2011) *Designing for the digital age: How to create human-centered products and services*. Wiley.

Finally, it is worth reading Gilbert Cockton's "Usability Evaluation" published online by the Interaction Design Foundation.

http://interaction-design.org/encyclopedia/usability_evaluation.html

13.9 Exercises

13.1 Design a usability test for comparing two makes of mobile device (such as a smartphone), using at least two of the ways of measuring usability described above. You should include details of how many participants you would use, and an explanation of why you chose your selected ways of measuring usability.

13.2 Design a study to evaluate the usability of an advanced photocopier/printer, using at least two of the ways of measuring usability that were described above. You should include details of how many participants you would use, and an explanation of why you chose your selected ways of measuring usability.

13.3 Design a usability test for evaluating the web site of an on-line retailer, using at least two of the ways of measuring usability that were described above. You should include details of how many participants you would use, and an explanation of why you chose your selected ways of measuring usability.

13.4 Summarize the evaluations in Exercises 13.1–13.3, comparing and contrasting how the devices being evaluated influence the number of participants and the choice of evaluation methods.

References

Berg, M. (1997). *Rationalizing medical work: Decision support techniques and medical practices*. Cambridge, MA: MIT Press.

Brooke, J. (1996). SUS: A ‘quick and dirty’ usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). London: Taylor & Francis.

Brown, C. M. L. (1988). *Human-computer interface design guidelines*. Norwood, NJ: Ablex.

Calfee, R. C. (1985). *Experimental methods in psychology*. New York, NY: Holt, Rinehart and Winston.

Campbell, D. T., & Stanley, J. C. (1963). *Experimental and quasi-experimental designs for research*. Boston, MA: Houghton Mifflin.

Card, S. K., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. Cambridge, MA: MIT Press.

Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems, 10*, 181–212.

Churchill, E. F., Nelson, L., Denoue, L., Murphy, P., & Helfman, J. I. (2003). The Plasma poster network: Social hypermedia on public display. In K. O’Hara, M. Perry, E. Churchill, & D. Russell (Eds.), *Social and interactional aspects of shared display technologies*. London: Kluwer Academic Publishers.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37*(1), 32–64.

Ericsson, K. A., & Simon, H. A. (1980). Protocol analysis: Verbal reports as data. *Psychological Review, 87*, 215–251.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (2nd ed.). Cambridge, MA: MIT Press.

Feurzeig, W., & Ritter, F. (1988). Understanding reflective problem solving. In J. Psotka, L. D. Massey, & S. A. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned*. Hillsdale, NJ: Erlbaum.

Goodman, E., Kuniavsky, M., & Moed, A. (2012). *Observing the user experience: A practitioner’s guide to user research* (2nd ed.). Waltham, MA: Morgan Kaufmann.

Hart, S. G., & Staveland, L. E. (1988). Development of the NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139–185). Amsterdam: North Holland.

Hock, R. R. (2002). *Forty studies that changed psychology*. Upper Saddle River, NJ: Prentice Hall.

Howell, D. C. (2010). *Statistical methods for psychology* (7th ed.). Belmont, CA: Wadsworth.

Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI—recording user input from interfaces under Windows and Mac OS X. *Behavior Research Methods, 38*(4), 656–659.

LaPiere, R. T. (1934). Attitude versus action. *Social Forces, 13*, 230–237.

Lazar, J., Feng, J. H., & Hochheiser, H. (2010). *Research methods in human-computer interaction*. New York, NY: Wiley.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2013). *Qualitative data analysis: A methods sourcebook*. Thousand Oaks, CA: Sage.

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM, 33*(3), 338–348.

Monk, A., Wright, P., Haber, J., & Davenport, L. (1993). Apendix 1—cooperative evaluation: A run-time guide. In *Improving your human-computer interface: A practical technique*. New York: Prentice-Hall.

Monk, A. F. (1998). Lightweight techniques to encourage innovative user interface design. In L. Wood (Ed.), *User interface design: Bridging the gap between user requirements and design* (pp. 109–129). Boca Raton, FL: CRC Press.

Mosier, J. N., & Smith, S. L. (1986). Application of guidelines for designing user interface software. *Behaviour and Information Technology, 5*, 39–46.

Müller, M. J., Haslwanter, J. H., & Dayton, T. (1997). Participatory practices in the software lifecycle. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., pp. 255–297). Amsterdam, NL: Elsevier.

NASA. (1987). *NASA Task Load Index (TLX) V 1.0. Users Manual*. Retrieved 10 March 2014, from http://humansystems.arc.nasa.gov/groups/TLX/downloads/TLX_comp_manual.pdf.

Navalpakkam, V., & Churchill, E. F. (in press). Eyetracking: A brief introduction. In J. S. Olson & W. Kellogg (Eds.), *Ways of knowing, HCI methods*. Heidelberg, Germany: Springer.

Nielsen, J. (1993). *Usability engineering*. Chestnut Hill, MA: AP Professional Press.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of CHI 90* (pp. 249–256). New York: ACM.

Nielsen, J., & Pernice, K. (2010). *Eyetracking web usability*. Berkeley, CA: New Riders.

Ray, W. J. (2008). *Methods toward a science of behavior and experience* (9th ed.). Belmont, CA: Wadsworth Publishing.

Reeves, B., & Nass, C. (1996). *The media equation: How people treat computers, television, and new media like real people and places*. New York: NY Cambridge University Press.

Ritter, F. E., Kim, J. W., & Morgan, J. H. (2009). *Running behavioral experiments with human participants: A practical guide* (Tech. Report No. 2009-1). Applied Cognitive Science Lab: The Pennsylvania State University, College of Information Sciences and Technology.

Ritter, F. E., Kim, J. W., Morgan, J. H., & Carlson, R. A. (2013). *Running behavioral studies with human participants: A practical guide*. Thousand Oaks, CA: Sage.

Smallman, H. S., & St. John, M. (2005). Naïve realism: Misplaced faith in the utility of realistic displays. *Ergonomics in Design, 13*(Summer), 6–13.

Todd, Z. (Ed.). (2004). *Mixing methods in psychology: The integration of qualitative and quantitative methods in theory and practice*. Abingdon, UK: Psychology Press.

Tullis, T., & Albert, B. (2008). *Measuring the user experience*. Burlington, MA: Morgan Kaufmann.

Whiteside, J., Jones, S., Levy, P. S., & Wixon, D. (1985). User performance with command, menu, and iconic interfaces. In *Proceedings of CHI'85 Human Factors in Computing Systems* (185–191). New York: ACM.

Woods, D. D., & Dekker, S. W. A. (2000). Anticipating the effects of technological change: A new era of dynamics for human factors. *Theoretical Issues in Ergonomic Science, 1*(3), 272–282.