CHAPTER

Usability testing 10

10.1 INTRODUCTION

Usability testing is often known as "user research." Often, in usability testing, we're not researching the user, we're researching the interface. We're trying to figure out how to make a specific interface better. However, user research is also a broader term that may include elements of design and development, such as personas, user profiles, card sorting, and competitive research that generally might not be considered "research" by those who consider themselves researchers (Kuniavsky, 2003). Furthermore, usability testing as a research method (utilizing representative users and representative tasks) can be used to learn more about how people interact with interfaces, even when the goal is not fixing the interface, but instead learning more about users and interactions. So, in usability testing, maybe we *are* researching the user?

10.2 WHAT IS USABILITY TESTING?

Usability testing, in general, involves representative users attempting representative tasks in representative environments, on early prototypes or working versions of computer interfaces (Lewis, 2006). If that sounds like a very broad definition, it is meant to be that way. The world of usability testing includes:

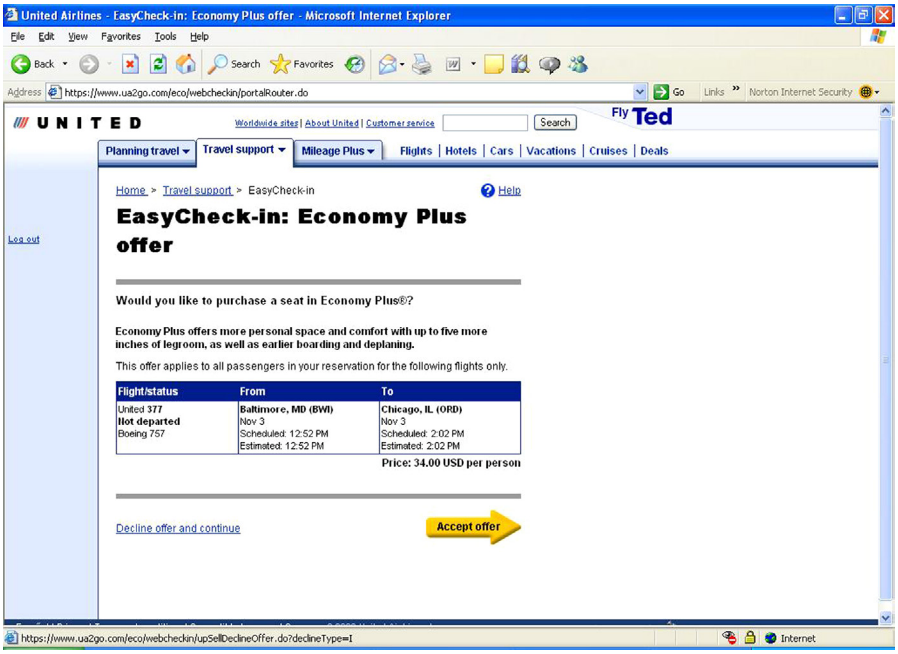
- testing prototypes that have only been built on paper (known as paper prototypes);
- testing screen mock-ups or wireframes which have no functionality
- testing screen layouts which have partial functionality
- testing prototypes that look complete but have a human behind the scenes responding (known as the "Wizard of Oz" technique);
- testing working versions of software before it is officially released;
- testing software that has already been implemented in existing systems.

The interfaces being usability tested are typically screen layouts for desktop, laptop, or tablet computers, as well as smart phones and other mobile devices. Usability testing can also be done to evaluate physical interaction with devices. Mobile devices frequently need usability testing, since the interaction approaches (such as multi-touch screens) are newer, more content is stuffed into a smaller screen size, and it can

Research Methods in Human-Computer Interaction. http://dx.doi.org/10.1016/B978-0-12-805390-4.00010-8

be easy to activate features by accident (e.g. holding the smartphone in your hand and hitting a button; or making a call, putting the phone up to your face, and accidentally selecting a feature).

All of these approaches to usability testing have one basic goal: to improve the quality of an interface by finding flaws-areas of the interface that need improvement. While usability testing should discover interface flaws that cause problems for users, at the same time, we want to discover what is working well with an interface design, so that we make sure to keep those features in place! What's an interface flaw? It is some aspect, some component, some widget of the interface that is confusing, misleading, or generally suboptimal. It is not about style or color preferences. Someone may prefer dark blue text instead of black text, on a white background and that's fine. It becomes a usability problem only when you have white, yellow, orange, or red text on a white background, all of which are hard for the eye to perceive. When we talk about usability testing, we are talking about discovering interface flaws that cause problems for a majority of people. Figure 10.1 gives an example of an interface that has a major flaw. The screen shot shows the process of checking in online for an airline flight. Once you enter your information, the website asks if you would like to upgrade your seat to the class called "economy plus." Typically, most individuals would not want to upgrade. However, the user's eye naturally goes to the large yellow arrow


United Airlines - EasyCheck-in: Economy Plus offer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Norton Internet Security
UNITED
Worldwide sites | About United | Customer service
Search
Fly Ted
Planning travel
Travel support
Mileage Plus
Flights | Hotels | Cars | Vacations | Cruises | Deals
Home > Travel support > EasyCheck-in
Help
Log out
Accept offer
Decline offer and continue
Internet

https://www.ua2go.com/eco/webcheckin/portalRouter.do

EasyCheck-in: Economy Plus offer

Would you like to purchase a seat in Economy Plus®?

Economy Plus offers more personal space and comfort with up to five more inches of legroom, as well as earlier boarding and deplaning.

This offer applies to all passengers in your reservation for the following flights only.

Flight/status	From	To
United 377 Not departed Boeing 757	Baltimore, MD (BWI) Nov 3 Scheduled: 12:52 PM Estimated: 12:52 PM	Chicago, IL (ORD) Nov 3 Scheduled: 2:02 PM Estimated: 2:02 PM

Price: 34.00 USD per person

https://www.ua2go.com/eco/webcheckin/upSellDeclineOffer.do?declineType=I

FIGURE 10.1

An airline check-in screen with at least one clear usability flaw.

Source: www.ua2go.com.

on the right, which would seem to continue to the next screen. In reality, clicking the yellow arrow causes the user to upgrade their seat. To continue without an upgrade, the user needs to click on the textual link on the left, which is small (in comparison to the arrow) and not obvious. This is a confusing and potentially misleading interface flaw (whether it was intentionally misleading is a question that we will not address). This is a very minor flaw to change. However, it will have a major improvement on user interaction and performance.

The range of usability testing is quite broad. Usability testing can involve hundreds of users, have a number of controls, and use a true experimental design. Usability testing can also involve a researcher sitting down next to three users, watching them go through the interface, and then taking basic notes on where the problems are. While both of these exercises can be called usability testing, it is more likely that the former would be considered research and would be published. Usability testing can involve hypothesis testing, tight controls, control groups, and a large enough number of participants to determine statistically significant differences. However, that’s not the way that most usability testing happens (Rubin and Chisnell, 2008). Why? In industry, the extra time needed to plan controls and do random assignments, and the high number of participants needed, are often a barrier to entry (Rubin and Chisnell, 2008). If the choice is that you must do all of those or nothing at all, businesses often choose to do nothing. Therefore, more flexible, easier, and quicker methods are often used. Where does usability testing end and research begin? It’s an unclear, fuzzy line and the distinction is not all that important.

10.3 HOW DOES USABILITY TESTING RELATE TO “TRADITIONAL” RESEARCH?

Usability testing can be considered a close cousin of traditional research methods, and is often known as “user research.” In reality, the approaches utilized in usability testing are often the same as those used in classic research. Metrics utilized in usability testing include measurement of task performance and time performance, similar to experimental design. Methods utilized as part of usability testing include surveys to measure user satisfaction. Observation techniques, from ethnography, are often utilized in usability testing. Key logging and clickstream analysis (see Chapters 12 and 13) can be utilized in usability testing. As discussed in other chapters of this book (primarily Chapters 12 and 13), there are many automated data collection methods that could be used for usability testing. In usability testing, the rights that participants have are the same as in any other type of research. The names of the participants must remain anonymous, participants must be informed of their rights and sign some type of informed consent form, and participants have the right to leave the research at any time, just as in traditional research.

However, usability testing often has different end goals. Usability testing is primarily an industrial approach to improving user interfaces. As an industrial approach, there is little concern for using only one research method or having strict controls. In fact, Wixon goes as far as to say that usability testing has more in common with engineering than traditional research (Wixon, 2003). Wixon's assertion is that usability testing, like engineering, is involved in building a successful product, in the shortest amount of time, using the fewest resources, with the fewest risks, while optimizing trade-offs. Often in industry, schedule and resource issues, rather than theoretical discussions of methodology, drive the development process (Wixon, 2003). One practice that is somewhat accepted in usability testing is to modify the interface after every user test, when major flaws are discovered, to help immediately eliminate the flaws and improve the interface. Making immediate changes to the interface allows for those changes to be evaluated during the next user test, which can help ensure that no new interface problems have been introduced in making the changes (Wixon, 2003). While this may not happen due to time constraints, it is an acceptable practice. Clearly, this practice would be considered unacceptable in experimental design, where the goal would be to ensure that all users in a group have the same treatment. And since usability testing is an industrial, practical approach, it is also important to note that not all interface flaws discovered during usability testing are fixed. Very often, the list of interface flaws discovered is prioritized and only the most serious problems are fixed.

By now, it should be clear that the goal of usability testing is to be practical and have a major impact. Since the goal is often to improve interfaces and have an important impact on the financial bottom line of a company, many companies don't publish their usability test findings, as they consider it confidential and a part of their competitive advantage. There are, however, a number of documented cases of usability testing that we have included in this chapter.

There are some similarities and some differences between usability testing, and the ethnography and participatory design methods discussed in chapter 9. Ethnography is more focused on understanding people, their groups, their processes, and their beliefs. Often, ethnographic methods are used as part of a systems development method called *participatory design* (again, discussed in detail in Chapter 9). The end goal of ethnography is simply understanding a group, an organization, or a problem, whereas the end goal of participatory design is building a computer system. Usability testing follows a similar pattern, with an end goal of improved interface design in a specific system. In fact, participatory design includes the stages of both ethnographic observation (in the user's situational context) and usability testing. Development methods or lifecycles, such as participatory design, the systems development lifecycle, the web development lifecycle, or community-centered design, can be thought of as recipes, with the individual activities, such as ethnographic observation and usability testing, as the ingredients in those recipes. The methods used in usability testing borrow most closely from experimental design and ethnography. Table 10.1 provides a comparison of classical research methods (such as experimental design and ethnography) and usability testing. Again, it is important to note that while many of the same approaches from classical research can be utilized in usability testing, they are often implemented differently, with different end goals.

To make things a bit more confusing, there is also research about usability testing! That is, research exists on evaluating which usability testing methods are

Table 10.1 Differences Between Classical Research and Usability Testing

Classical Research Source	Classical Research Description	Usability Testing Description
Experimental design	Isolate and understand specific phenomena, with the goal of generalization to other problems	Find and fix flaws in a specific interface, no goal of generalization
Experimental design	A larger number of participants is required	A small number of participants can be utilized
Ethnography	Observe to understand the context of people, groups, and organizations	Observe to understand where in the interface users are having problems
Ethnography	Deep participatory embedding of the researcher in the community is often encouraged	Researcher participation is not encouraged, except when an intervention is needed to help the user get "unstuck" (with strict protocols for doing so)
Ethnography	Longer-term research method	Short-term research method
Ethnography and experimental design	Used to understand problems or answer research questions	Used in systems and interface development
Ethnography and experimental design	Used in earlier stages, often separate from (or only partially related to) the interface development process	Can take place as early as paper prototypes, where there is more potential impact on the interface, but often takes place in later stages, after interfaces (or prototype versions of interfaces) have been developed, with less potential impact on the interface
Ethnography and experimental design	Used for understanding problems	Used for evaluating solutions

most effective. For instance, in the debate on how many participants you need (see Section 10.5.3), the focus is not on improving specific interfaces, but on understanding and improving the usability methods themselves. But that isn't usability testing, that's research on how to do usability testing and that's a whole different topic!

10.4 TYPES OF USABILITY TESTING OR USABILITY INSPECTIONS

There are many different types of usability testing. A more general term, "usability engineering," has sometimes been used to describe any process or activity that aims to improve the ease of use of an interface. Under this heading, and sometimes under the heading of usability testing, there are three distinct categories: expert-based testing, automated testing, and user-based testing.

An expert-based test involves interface experts in using a number of different structured methods for finding interface flaws. An automated test is a software program that applies a series of guidelines (developed by the experts) to an interface and determines where the interface doesn't meet the guidelines. A user-based test involves representative users performing representative tasks (at various stages in the development process). While user-based tests are the majority focus of usability evaluation, expert-based tests and automated tests are sometimes used in human-computer interaction (HCI) practice.

As multimethod research approaches gain strength, we expect to see a greater appearance of expert and automated usability testing. Note that expert and automated usability tests are sometimes known as *usability inspections,* and *usability testing* is reserved for user-based testing. Whole books have been written about each type of usability testing, so this chapter provides only a summary of each type. Since the primary interest in HCI research is users and collecting data from users, this chapter primarily focuses on user-based testing. First, we briefly discuss expert-based testing and automated testing.

10.4.1 EXPERT-BASED TESTING

Expert-based tests are essentially structured inspections by interface experts. The people who developed the prototype interface being evaluated should not be involved with the expert review, as that may bias the results. People who are unfamiliar with the interface should carry out the expert reviews. Expert-based tests are often used in conjunction with user-based tests, but the expert-based tests always come first. Interface experts are experts in interfaces but they are typically not experts in the tasks to be performed within a certain interface. Conversely, representative users are typically experts in performing the tasks but are not experts in interface design. Often a certain portion of interface functionality can be understood and improved without a deep understanding of the tasks, but other portions of the interface can only be examined with a deep understanding of the tasks involved.

Interface experts first use a structured inspection to attempt to uncover some of the more obvious interface flaws, such as confusing wording, inconsistent or misleading layouts, and color inconsistency. If possible, suggested improvements to the interface from the expert review should be made before user-based usability testing occurs. This timeline allows the experts to find the obvious interface flaws and get them fixed; the users can then find the deeper, more granular, and task-related interface flaws which may not be obvious to the interface experts (Lazar, 2006). If there are many interface flaws and no expert has reviewed the interface, the users may be distracted by the major interface flaws and may be unable to help the developers by identifying the more granular, task-based flaws.

There are a number of different types of expert review, also known as expert inspections or usability inspections. The most common expert reviews are the heuristic review, the consistency inspection, and the cognitive walkthrough. In a heuristic review, an expert takes a set of heuristics (rules of thumb) and compares the heuristics to the interface in question. Heuristics are short sets of usually no more than 10 interface rules. To be truly effective, the expert must be very familiar with the heuristics and have previous experience in interpreting them. Lazar provides a list of various sets of heuristics for different types of websites (Lazar, 2006) but the best-known set of broad interface heuristics is probably Shneiderman's 8 Golden Rules of Interface Design (see Table 10.2).

Table 10.2 Shneiderman's 8 Golden Rules of Interface Design

Strive for consistency
Cater to universal usability
Offer informative feedback
Design dialogs to yield closure
Prevent errors
Permit easy reversal of actions
Support internal locus of control
Reduce short-term memory load

(Shneiderman et al., 2017)

In a consistency inspection, one or more experts review a series of screens or web pages for issues of consistency in layout, color, terminology, or language. Sometimes, an organization has a specific set of style guidelines (for colors and typefaces) and a consistency inspection can check for overall consistency with those style guidelines.

A cognitive walkthrough is an expert review method in which interface experts simulate users, "walking through" a series of tasks. The experts must have experience with general interface design and a good understanding of who the users are and what tasks they are expected to perform in the interface that is being evaluated. Because of the exploratory nature of a cognitive walkthrough, it can give an understanding of how users might interact with an interface the first time that they attempt to use it (Hollingsed and Novick, 2007). Both high-frequency tasks and rarely occurring but important tasks (such as error recovery) should be included in a cognitive walkthrough (Shneiderman et al., 2017). Because it is task-based, rather than rule-based for experts, it is still somewhat controversial, as some people feel that it is not as productive as user-based testing.

Not as popular as the previous three methods, but still occurring often, is the guidelines review, in which an expert compares a set of interfaces to a previously written set of interface guidelines. While this sounds like a heuristic review, the main difference is that a guidelines review uses a large set of guidelines (usually 10–200). Heuristic reviews take place more often because they are easier and take less time. However, guideline reviews are more thorough. Probably one of the best-known sets of guidelines is the Web Content Accessibility Guidelines (WCAG, currently in version 2.0), created by the World Wide Web Consortium (http://www.w3.org/WAI). These guideline documents provide guidance on making website content accessible for people with disabilities. Internationally, most laws that deal with accessible web content were written based on the WCAG. The Web Accessibility Initiative also has guidelines related to authoring tool accessibility, user agent accessibility, and rich Internet application accessibility. These guidelines, while being commonly used, can be overwhelming in scope and so the Web Accessibility Initiative also offers shorter versions of the guidelines documents (such as checkpoints and quick tips) which can be considered as heuristics. Other commonly used guidelines include the operating systems interface guidelines documents from Apple and Microsoft, the research-based web design and usability guidelines from the US government and the KDE or GNOME interface guidelines. In addition, firms such as the Nielsen Norman Group have large numbers of specialized guideline sets that are available for a price.

Other types of expert review, such as the formal usability inspection and the pluralistic walkthrough, are not as common (Hollingsed and Novick, 2007). If you are interested in different types of expert review, you should read the classic book on expert reviews (Nielsen and Mack, 1994) or recent HCI papers about expert review methods. However, since expert-based reviews really don't involve users, we won't go into any more details on this topic.

10.4.2 AUTOMATED USABILITY TESTING

An automated usability test is a software application that inspects a series of interfaces to assess the level of usability. Often, this works by using a set of interface guidelines (described in Section 10.4.1) and having the software compare the guidelines to the interfaces. A summary report is then provided by the automated usability testing application. Automated usability testing applications are often used when a large number of interfaces need to be examined and little time is available to do human-based reviews. The major strength is that these applications can read through code very quickly, looking for usability problems that can be picked up. These applications typically have features to either offer advice about how the code should be fixed or actually fix the code. However, the major weakness is that many aspects of usability cannot be discovered by automated means, such as appropriate wording, labels, and layout. And most automated tools are designed only to test web interfaces. For instance, an application can determine if a web page has alternative code for a graphic (important for accessibility, and a requirement under the WCAG 2.0), by examining to determine the existence of an <alt> attribute in an <img> tag. However, an application cannot determine if that alternative text is clear and useful (e.g. "picture here" would not be an appropriate text but it would meet the requirements of the automated usability testing application). In many situations like that, manual checks are required. A manual check is when one of these applications notes that because of the presence of certain interface features, a human inspection is required to determine if a guideline is complied with (e.g. if a form has proper labels).

Automated usability testing applications are good at measuring certain statistics, such as the number of fonts used, the average font size, the average size of clickable buttons, the deepest level of menus, and the average loading time of graphics (Au et al., 2008). These are useful metrics, but they do not ascertain how users interact with those interfaces, only how well these interfaces comply with some guidelines. Automated tools can also help with determining a high-level view of thousands of web pages, for example, within an organization, to determine how many are meeting certainly basic usability requirements. For instance, Lazar et al. (2017) utilized automated accessibility testing tools to examine which US federal agencies had accessibility features present on a large portion of their web sites (not whether a specific web page was fully compliant or not). Automated tools are good for tasks such as that, determining the presence of features and getting a high-level overview. A large number of tools exist for automated accessibility testing, including standalone applications such as Deque WorldSpace, Cryptzone ComplianceSherriff, and SSB Accessibility Management Platform, as well as free web-based tools such as A-Checker, WAVE, and Functional Accessibility Evaluator, all of which check interfaces for compliance with the WCAG 2.0 guidelines. A classic article about the concepts behind automated usability testing can be found in the ACM Computing Surveys (Ivory and Hearst, 2001).

10.5 THE PROCESS OF USER-BASED TESTING

User-based testing is what most people mean when they refer to usability testing. Mostly, it means a group of representative users attempting a set of representative tasks. This can take place very early in development, during development, or very late in development. It is better to start doing user-based testing earlier rather than later, when the results can influence the design more and when costs to make changes are much lower. Ideally, user-based testing would take place during all stages of development, but that is not always possible. Why do we do usability testing? As much as designers try to build interfaces that match the needs of the users, the designers are not users and even the users themselves sometimes cannot clearly identify their interface needs. So interface prototypes, at various stages, need to be tested by users. Note that users are testing interfaces, but users are not being tested. This is an important distinction. Furthermore, some authors even go so far as to say that the developers who create an interface design should not be the ones who moderate a usability test (Rubin and Chisnell, 2008). If you create an interface, you are likely to be supportive of that interface, feel that you have time invested in it, and may not be as open to user suggestions. From a strict experimental point of view, the interface developer shouldn’t moderate a usability test or interact with the participants (although the developer can observe the testing to learn what aspects of their design aren’t working well). However, since perfect design isn’t the goal of usability testing, there are situations where the interface developer serves double duty and moderates the usability test.

10.5.1 FORMATIVE AND SUMMATIVE USABILITY TESTING

Usability testing that takes place early in development tends to be exploratory and to test early design concepts. Sometimes, this is known as formative testing and may include wireframes or paper prototypes, also known as low-fidelity prototypes (Dumas and Fox, 2007). This type of usability testing is often more informal, with more communication between test moderators and participants (Rubin and Chisnell, 2008). In early exploratory testing, there is more of a focus on how the user perceives an interface component rather than on how well the user completes a task (Rubin and Chisnell, 2008). Paper prototypes are especially useful, because they are low cost and multiple designs can be quickly presented and evaluated by participants. In addition, because paper prototypes involve little development time, designers and developers tend not to become committed to a specific design early on. And users may feel more comfortable giving feedback or criticizing the interface when they see that not much work has been done yet on the interface. With fully functional prototypes, users may be hesitant to criticize, since they feel that the system is already finished and their feedback won't matter that much. More information on paper prototyping can be found in Snyder (2003).

Usability testing that takes place when there is a more formal prototype ready, when high-level design choices have already been made, is known as a summative test. The goal is to evaluate the effectiveness of specific design choices. These mostly functional prototypes are also known as high-fidelity prototypes (Dumas and Fox, 2007).

Finally, a usability test sometimes takes place right before an interface is released to the general user population. In this type of test, known as a validation test, the new interface is compared to a set of benchmarks for other interfaces. The goal is to ensure that, for instance, 90% of users can complete each task within 1 minute (if that statistic is an important benchmark). Validation testing is far less common than formative or summative testing.

It is important to note that there are variations in how usability testing is structured, regardless of the type of usability test or the stage of interface development. So in general, the data collected in a validation test or summative test will tend to be much more quantitative, and less focused on users "thinking aloud." More formative testing, on earlier prototypes, will tend to be more thinking aloud and qualitative data. But none of these are 100% definite. With well-developed paper prototypes, you theoretically could measure task performance quantitatively, and you could utilize the thinking aloud protocol when an interface is fully developed. The key thing to remember is that, the more that users "think aloud" and speak, the more that their cognitive flow will be interrupted, and the longer time a task will take to complete (Hertzum, 2016; Van Den Haak et al., 2003). It is also important to remember that, at first, individual children participants involved in usability testing may not feel comfortable criticizing an interface out loud (Hourcade, 2007), but pairs of children doing usability testing may be more effective (Als et al., 2005). Usability testing is flexible and needs to be structured around the activities that are most likely to result in actual changes in the interface being evaluated.

Different authors use different definitions for these terms. For instance, we have used the definitions from Rubin and Chisnell. West and Lehman, however, define formative tests as those that find specific interface problems to fix and summative tests as those that have a goal of benchmarking an interface's usability to other similar interfaces (West and Lehman, 2006). Sauro and Lewis (2012) have a similar view, describing any type of usability test to find and fix usability problems as formative, and describe summative only as metrics for describing usability.

The one thing that most authors agree on is that earlier, formative usability tests tend to focus more on qualitative feedback, moderator observation, and problem discovery, whereas summative usability tests tend to focus more on task-level measurements, metrics, and quantitative measurements (Lewis, 2006). The "Usability Testing of the Kodak Website" sidebar gives an example of formative and summative usability testing.

USABILITY TESTING OF THE KODAK WEBSITE

The Eastman Kodak Company is one of the world's largest manufacturers and marketers of imaging products. Both formative and summative usability testing took place on the Kodak website.

Formative testing took place on a paper prototype of the new home page design, specifically the links and groups. Twenty participants were given 30 tasks and were asked to identify the homepage link most likely to lead to the information that would complete that task. Participants were then asked to describe what type of content they expected to find behind each homepage link. Finally, participants were given descriptions of what actually was behind each home page link, and were asked to rate how well the label matched the actual content.

Later, summative testing with 33 participants took place on a working prototype of the new home page and all top-level pages on the site. A list of 22 tasks was developed, but each participant was given only 10 information-seeking tasks to complete. Some tasks were attempted by all 33 participants, while other tasks were attempted by only 11 participants. All links were functional, although not all visual design elements on the pages were complete. Each participant was given a maximum of 3 minutes to complete each task. Task completion for individual tasks ranged from 100% to 9% in the allotted 3 minutes. Based on the results of the usability testing, changes were made to the pages, including removing images along the left side of the page, adding longer descriptors to more of the links, and labeling major chunks of information (Lazar, 2006).

Whether a usability test is formative, summative, or validation can influence how formal or informal the usability test is. At one end of the spectrum is a formal approach to usability testing, which parallels experimental design. This form of usability testing can involve specific research questions, research design (between-subject design or within-subject design), and multiple interfaces to test. If you are using inferential statistics, hypotheses, a control group, large numbers of subjects, and strict controls on user recruitment, usability testing may, in fact, become experimental design. The only difference would be that experimental design is looking for statistically significant differences between groups to learn some research truth, whereas usability testing is looking for ways to identify usability flaws and improve specific interfaces.

10.5.2 STAGES OF USABILITY TESTING

Usability testing is not something that just happens. It requires a lot of advance planning. Different authors on the topic describe different steps, but the reality is that there are a lot of advance planning steps involved. See Table 10.3 for examples of the stages of usability testing from two different authors.

Table 10.3 Stages of Usability Testing From Different Authors

Stages of Usability Testing	
(Rubin and Chisnell, 2008)	(Lazar, 2006)
Develop the test plan	Select representative users
Setup the test environment	Select the setting
Find and select participants	Decide what tasks users should perform
Prepare test materials	Decide what type of data to collect
Conduct the test sessions	Before the test session (informed consent, etc.)
Debrief the participants	During the test session
Analyze data and observations	Debriefing after the session
Report findings and recommendations	Summarize results and suggest improvements

There are a number of stages of usability testing that seem very similar to experimental design (see Chapter 3). Often, a usability expert, taking the role of the usability moderator, manages the process. For more detailed information about moderator roles, we suggest that you consult Dumas and Loring (2008). The moderator should determine which users would be appropriate, representative participants to take part in the usability testing. If the typical users of the new interface system are nurses at a hospital, it is inappropriate (and probably unethical) to use undergraduate students in business to perform the usability testing (although nursing students might be appropriate, depending on the level of domain knowledge and job experience required). If appropriate user-centered design methods have been utilized, there should be existing user personas and task scenarios that can help guide you in this process. Some of the most common criteria for determining representativeness of users, include age, gender, education, job responsibility and or/domain expertise, technical experience (in general), and experience with specific software or hardware devices (Tullis and Albert, 2008).

Once you have figured out who the representative, appropriate users are, the next goal is to try and recruit them. Again, this is very similar to experimental design. For instance, users expect to be paid for their participation in usability testing, just as they expect to be paid for their participation in an experimental study. However, recruitment in usability testing is generally seen to be more flexible than in experimental design, and samples of convenience are common and appropriate (Tullis and Albert, 2008). While it is very important that the recruited participants accurately represent the target user population, it is less relevant how you recruit those users. Unless you are dealing with multiple user populations across cultures, countries, or languages (in which case you may want to do usability testing at each site), it can be satisfactory, for instance, to recruit users from only one or two companies or in only one geographic area.

10.5.3 HOW MANY USERS ARE SUFFICIENT?

One of the most common questions when planning usability testing is "how many users do I need to have?" It's also a bit of a hotbed of discussion in the HCI community, and a consensus has not emerged over time. If you were doing a strict experimental design, the types of research design and the statistical tests that you run would dictate the minimum number of participants required. However, usability testing has different goals and different requirements.

Many people say that five users is sufficient, and that five users will find approximately 80% of usability problems in an interface (Virzi, 1992). This has become an often-quoted number in HCI, but many other researchers disagree with the assertion. The major challenge in determining the right number of users, is that you don't know in advance how many interface flaws exist, so any estimate of how many users are needed to find a certain percentage of interface flaws is based on the assumption that you know how many flaws exist, which you probably don't. Other research studies have found that five users are not sufficient to discover and identify a majority of usability flaws (Lindgaard and Chattraticharat, 2007; Spool and Schroeder, 2001). In a classic paper, Nielsen and Landauer, who in earlier work had asserted the number five, expressed that the appropriate number depends on the size of the project, with seven users being optimal in a small project and 15 users being optimal in a medium-to-large project (Nielsen and Landauer, 1993). However, in that same paper, they indicated that the highest ratio of benefits to costs is when you have 3.2 users doing usability testing (Nielsen and Landauer, 1993). In an analysis of existing research on the topic, Hwang and Salvendy (2010) suggest that 10 ± 2 is the optimal number of users for usability testing, although more recent work by Schmettow (2012) suggests that even 10 users is not enough to discover 80% of the usability problems.

Lewis says that all authors could theoretically be right about the appropriate number of users, as it depends on how accurate they need to be, what their problem discovery goals are, and how many participants are available (Lewis, 2006). Even if five users are enough, what happens when you have multiple user groups taking part in usability testing. Do you need five users from each group? Lindgaard and Chattratichart (2007) take a different approach: they assert that the number of usability flaws found depends more on the design and scope of the tasks, rather than on the number of users.

By now, "five participants in usability evaluation" is part of the HCI lore, in the same way that "7 ± 2 menu items" is part of the HCI lore. We are told that we should organize our menu items and menu bars into chunks of five to nine items, based on classic psychological research literature (Miller, 1956). However, this is misleading: the 7 ± 2 limitation in short-term memory applies to recall, not recognition, and most interface design (including menus) is recognition, where we see or hear an icon or item and think, "oh yes, that's what I wanted" (Preece et al., 2002 explained this well, although their explanation hasn't appeared in later editions of their book). However, "five participants" and "7 ± 2 menu items" remain part of the HCI folklore, even when there is real debate about their validity.

The reality is that most usability testing will never uncover all, or even most, of the usability flaws. And even if all of the flaws were uncovered, most of them will never be fixed. Instead, the goal should be to find the major flaws, the flaws that will cause most problems, and get them fixed. From an industry point of view, the exercise of finding flaws, without the consideration of whether they can be fixed, is not of value (Wixon, 2003). It simply would not make sense to expend all of the available "usability time" in a development lifecycle on finding flaws, rather than balancing time between finding flaws and fixing flaws. It may be useful to examine the effectiveness of various usability testing methods. But in industry, usability testing logistics are often driven not by what should or needs to be done, but instead, on how much time is left in the development process, how much money has been set aside by management for usability testing, and how many users are available and willing to participate. For instance, in usability testing on the website of the American Speech-Language-Hearing Association, the usability engineer identified 16 different user populations for the website. But after the prototype of the new website was built, the budget only allowed for usability testing with school-based, speech-language pathologists, the largest group of users for the website (Lazar, 2006). So instead of saying, "how many users must you have?," maybe the correct question is "how many users can we afford?," "how many users can we get?" or "how many users do we have time for?"

10.5.4 LOCATIONS FOR USABILITY TESTING

Usability testing can take place anywhere. It can take place in a fixed laboratory, a workplace, a user's home, over the phone, or over the web. The location may be determined by what locations are available or where participants are, as well as what type of data you want to collect. None of the types of location are superior to any others. You should use whatever works for your specific usability testing project.

The most traditional setting for usability testing is a two-room setup. The user sits in one room and works on the tasks on a computer. Microphones and cameras record what the user is doing and output from the user's computer screen is also recorded. In the other room, the test moderators, and other stakeholders, sit and watch what the user is doing during the test. The moderators' room generally has a number of computer screens and monitors and the recording equipment, so all appropriate data can be recorded. In addition, the moderators' room often has a one-way mirror so that the moderators can directly observe what the user is doing, but the user cannot see into the moderators' room (see Figure 10.2). If a one-way mirror is not possible (either due to structural concerns or the moderators' room being located elsewhere in the building), a large image projected on to a wall is sufficient for the same purpose.

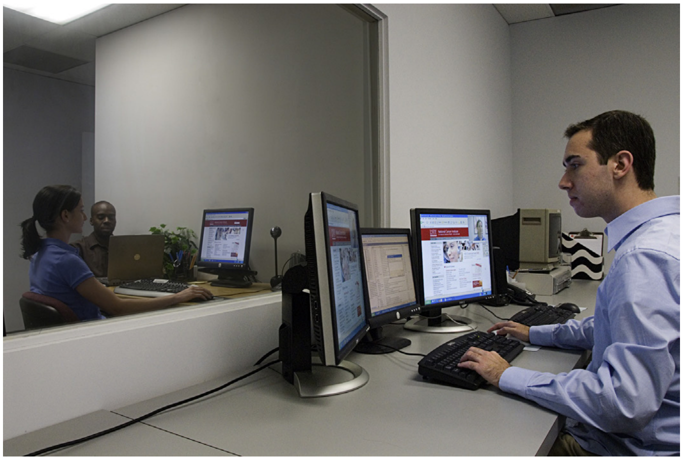


FIGURE 10.2

A formal usability laboratory with a one-way mirror.

Source: Photo by Elizabeth Buie for UserWorks, Inc., a usability consulting firm located in Silver Spring, MD, www.userworks.com.

While a formal usability laboratory is typically used for desktop or laptop computer applications, with minor modifications to the camera angles and mounting, a formal laboratory can also be utilized for usability testing of hand-held and mobile devices. For instance, one solution utilized for videotaping interactions on a mobile device is a document camera, which is often available in a classroom or presentation room. Readers are suggested to reference (Schusteritsch et al., 2007) on different types of camera mountings and logistics for usability testing of hand-held devices.

Figures 10.3 and 10.4 show two examples of formal, fixed usability labs. One lab layout is located at a university, where there is only one participant room. The other lab layout is from the US Census Bureau, where three participant rooms are connected to one evaluation room. It is important to note that while it is very good to have a formal usability laboratory, and the willingness to spend money and commit space may speak about the importance of usability to an organization, having a fixed usability laboratory is NOT necessary for usability testing.

Another possible location for usability testing is in the user's workplace or home. This may help in recruiting users, because it's less of a hassle for the users

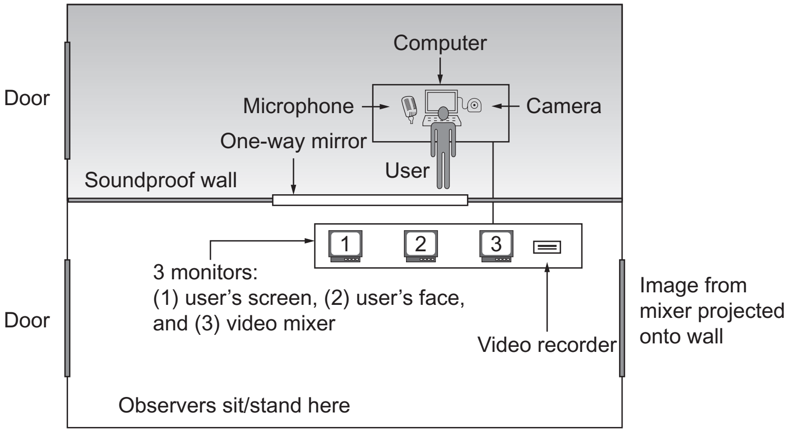


FIGURE 10.3

Potential layout for a single-participant usability lab.

From Lazar, J., 2006. Web Usability: A User-Centered Design Approach. Addison-Wesley, Boston.

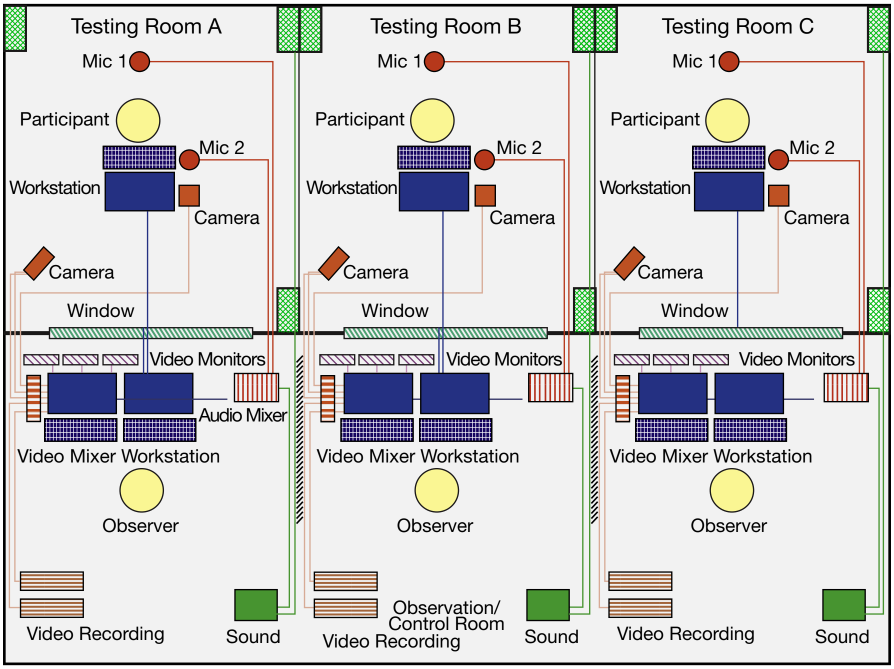


FIGURE 10.4

Potential layouts for a multiuser usability lab.

From US Census Bureau.

than having to go to a usability laboratory or a central location. Visiting users in their workplace or home may also be easier if you are working with users with disabilities, for whom transportation is often a challenge (see Chapter 16). In many ways, the user's workplace or home setting is ideal for usability testing. The user is exposed to customary space, noise, and attention limitations while testing the interface. The user may feel most comfortable in their normal environment, using their own technology, which again, may enhance their performance. In fact, testing in the user's natural setting goes along with the ideals of ethnography (see Chapter 9). For the user, it's the easiest and most natural form of usability testing. However, for the usability testing moderator, it can be the most challenging.

First of all, a lot of travel may be needed to visit each user. Secondly, a decision needs to be made: do you install the software or interface on the user's computer (which is more natural but may involve technical problems) or do you bring a laptop with the software or interface installed on it (which is technically easier but it's not the computer that the user is familiar with, so you may get some conflicting results). Chapter 16 provides an in-depth discussion of this decision. If you are usability testing a mobile device, you typically bring the device along. Thirdly, how are you going to record data? There are different approaches, all with benefits and drawbacks. If you observe the users by sitting beside them, this may make them feel uncomfortable and they may act differently. You can use a number of technical approaches, but they take some time to set up before you begin the session. For instance, you could use data logging (where user keystrokes are recorded), record audio or send screen output to another local computer. If you have the equipment, or the budget to rent equipment, you could use a portable usability laboratory. A portable usability laboratory includes the same equipment (cameras, microphones, digital recording devices, etc.) as a fixed usability laboratory, but in a portable case, with very long wires (or a wireless signal). The idea is that, when you get to the user's home or workplace, you set up the equipment so that it essentially mirrors the setup in a fixed lab, so that a camera and a microphone are trained on the user and screen capture is in place. You then find a location near the user (not next to the user, or where the user feels your presence, or where you can physically see the user) but where the equipment wires are long enough to reach (or wireless signals can help with this). You can both record audio/video and watch a live feed and take notes. This may be ideal from a research point of view, since you get rich data capture and recording and the user is in their most comfortable and familiar setting, but the downside is that portable usability equipment is very expensive, takes a long time to set up, and there are often technical problems. If you are usability testing a mobile device, how do you accurately observe or record user actions on a device that may be too small to watch, unless you are standing right behind the user? If they are continuously moving the device around their own environment (as most people do), how do you observe or record data, aside from data logging (Schusteritsch et al., 2007)?

Sometimes, it is not feasible to do usability testing in a centralized location at a usability lab or in a user's workplace or home. It could be that the representative user population is not within easy traveling distance of the usability evaluators or moderators. Or there could be other logistical factors that limit the ability to do face-to-face usability testing. For instance, it could be appropriate to do remote usability testing with individuals with disabilities for whom transportation might be a problem (see Chapter 16). When an interface will be deployed in multiple countries, it is necessary to have users from all of the countries test the interface but it may not be possible for the evaluators to visit all the countries (Dray and Siegel, 2004). For doing usability testing involving children, the familiarity with the testing location is only one of the potential logistical concerns (also including topics such as how to get consent for children to participate). For those who are doing usability testing involving children, it is suggested to consult a thorough guide to children and HCI, such as Fails et al. (2012) or Hourcade (2007).

Remote usability testing is typically where users are separated from the evaluators by space, time, or both (Andreasen et al., 2007). Video, audio, and network connections allow evaluators to monitor users, including streaming output from the user's screen and clickstream data. While this used to be done using videoconferencing and private networks (Dray and Siegel, 2004), now, it is just as likely that a web-based remote usability testing tool (such as UserZoom) is utilized for remote testing. One of the challenges with remote testing is the difficulty (with synchronous videoconferencing) or impossibility (with many web-based usability testing tools which offer limited video and audio) of picking up nonverbal and interpersonal cues. Also with asynchronous remote testing, it is hard (or impossible) to provide instructions when things "go wrong," you can't ask any probing questions, and you often miss the context of what was happening. To offset these drawbacks, there are many benefits of remote usability testing, such as easy access to more potential participants (since you are not geographically limited), and easy collection and analysis of clickstream data (which can easily be turned into graphs and heatmaps).

Remote usability testing, on the whole, works better for summative testing, when you are more interested in quantitative metrics, than for formative testing, where you tend to be more interested in the qualitative observations (Dray and Siegel, 2004). In addition, synchronous remote usability testing, where the evaluators are observing the users in different locations at the same time using videostreaming, may be more effective than asynchronous testing, where the evaluator observes at a later time (Andreasen et al., 2007). While remote usability testing can be technically very challenging and small problems can delay testing (since moderators aren't there to address any technical problems), it can be a very useful technique in the toolbox of the usability evaluator. Table 10.4 displays benefits and drawbacks of remote usability testing.

Finding an appropriate place to do usability testing, and recruiting a sufficient number of representative users, is always an ongoing challenge. The next two sidebars describe how two of the leading technical companies, Google and Yahoo, use innovative approaches for recruitment of participants for usability testing.

Table 10.4 Benefits and Drawbacks of Remote Usability Testing

Benefit	Drawback
Easy access to a greater number of participants Participants have more flexibility in participating in a usability test on their own schedule, and researchers can run multiple usability tests at the same time Easy collection and analysis of clickstream data Works better for summative testing, when you are collecting quantitative metrics	Difficult or impossible to pick up nonverbal and interpersonal cues Hard (or impossible) to provide instructions when things "go wrong" Researchers can't ask any probing questions based on what occurs Researchers often miss the context of what was happening

USER NIGHTS AT YAHOO! (WRITTEN BY GARY MOULTON)

Yahoo is unique in combining its UX Researchers and Accessibility specialists to form an organization called User Experience Research and Accessibility (UXRA). UXRA partners with product groups throughout the development life cycle to gather, analyze, and present observations from user research when teams are gathering requirements, have a fully developed idea, and are working on a new mobile app or Web property. Yahoo is unique in combining Accessibility specialists with traditional User Experience Researchers. They work closely together with individuals that identify themselves as having a disability to observe and quantify user interactions with, and validate the compatibility of, mobile apps and Web properties with popular assistive technologies (e.g. screen readers, alternate input devices, etc.).

Yahoo's UXRA uses a wide variety of qualitative and quantitative research tools and methodologies for providing research throughout the development lifecycle. One of the unique methods is called "User Night" where up to 100 external users are paired individually with members of a particular Yahoo product team, who are called "Yahoos." Yahoos are briefed in advance and provided a script and coaching to run their own study with their participant. For up to an hour, they have conversations about the use of their product and observe real-world use on the participants' own devices (phones and/or laptops). After the event, team members share key findings in large group settings, and findings from these sessions are aggregated and fed back to the entire team. This process enables rapid, larger-scale feedback than is possible to be obtained in a single-day, five-user, traditional usability study. These events create unique empathy among product team members for real users, their issues as well as joys, in using the product that they spend each day building.

(Continued)

USER NIGHTS AT YAHOO!—CONT'D

Yahoo conducts several User Nights per quarter, each focused on a particular product or product feature, like Yahoo Search, Yahoo Finance, or Yahoo Fantasy Sports. The product manager (PM) will partner with their embedded UX researcher to determine the goals of the event (e.g. user reaction to a Yahoo product or feature), the number, and demographics of the participants and outline the tasks users will be asked to complete during User Night.

The UXRA researcher works with the project manager before the User Night to create a script based on the task list. The scripts are very detailed and comprehensive including methods of setting user expectations, "do's and don'ts," reminders for the Yahoos conducting 1:1 interactions, and the step-by-step way in which the specific tasks are to be conducted. An accessibility specialist will also be consulted to determine if assistive technology users will need to be recruited. The featured product and the specific tasks to be covered must not have any significant accessibility issues (i.e., a screen reader user should be able to successfully complete the tasks) when individuals with disabilities will be recruited.

Following this initial meeting, those in UXRA tasked with recruiting begin to solicit users for participation. Recruiting is accomplished using multiple methods including emailing a list of individuals who have volunteered to be included in Yahoo's User Night database and making targeted calls to individual users. Some User Nights require a higher percentage of users who have used Yahoo products previously and others require those who have never used a Yahoo product or participated in a previous User Night. The number of participants recruited for a User Night averages from 50 to 100 depending on the study.

To encourage participation, User Nights are held in the evening from about 7:00 to 9:00 pm. Upon arrival, users register, sign NDAs (nondisclosure agreements) and are feted with "heavy" hors d'oeuvres or a light dinner. This serves several practical purposes: it ensures everyone is on time and ready when the study begins, socializes, and relaxes nervous participants, and provides time to prepare Yahoo product teams for their 1:1 interactions and coach them on best practices for observation.

One hour before the start of a Yahoo User Night, the UXRA researcher and PM conduct an orientation meeting with the Yahoos who will help facilitate the user testing. They review the User Night agenda and schedule (welcome, dinner, pairing with users, making observations, note-taking, debriefing, etc.), review the script and answer any last-minute questions. Due to the large number of product developers at Yahoo, User Night is often as new to the Yahoo product developers as it is to the user, so this employee orientation time is crucial in making the night successful and enjoyable.

User Night "officially" begins when a Yahoo is matched with a user. Yahoo's IT department provides all the equipment (e.g. smartphones or laptop computers) when a user's own device can't be used—typically due to using prerelease software that they're not allowed to install on their own devices. Yahoos conducting the session with the user are advised to mostly "watch and listen" and note any feedback or suggestions users provide for improving the product experience as they guide their assigned user through the script. This makes up the bulk of the time spent at User Night and lasts approximately 1 hour. These interactions become so engaging it's often difficult to bring the night to a close, but the night must come to an end.

When the dust settles, and the users have left, the most critical part of User Night takes place. All of the Yahoos remain to conduct a debriefing with the UX researcher, accessibility specialist, and PM to discuss what was learned. Commonly experienced issues and observations quickly rise to the surface, but it is also important to capture the odd, curious and unusual observations, and feedback as these provide opportunities for further investigation in the future. A report recording, prioritizing, and analyzing all these findings is later created and shared with the entire product team such that those who were not able to participate in the User Night can also benefit from what was observed.

Yahoo's User Night is a unique and innovative methodology to observe and interact with a large number of users in a very short time period, gather immediate, impactful feedback, and provide product groups with prioritized and actionable improvements that can be used to make Yahoo products, services, and technology more usable and accessible. It has proven to energize product teams, encourage deeper consideration of end users during design, and result in the delivery of improved products faster into market, with higher quality and greater customer satisfaction.

THE GOOGLE RESEARCH VAN (WRITTEN BY LAURA GRANKA)

The research van is the newest method in Google's User Experience (UX) toolkit, aiming to overcome some limitations of traditional lab-based UX research. We created the van to help improve Google's user research studies by enabling more participant diversity, agile recruiting, and flexibility when planning and executing research.

At Google, we do much of our research in UX labs at our headquarters in Mountain View, CA, and at our other major locations, like New York City, NY and Seattle, WA. As with all methods, research in physical onsite labs has its limitations, and in our case, we were growing increasingly concerned about participant diversity—namely routinely doing research with people willing and able to participate in our *onsite* research studies.

(Continued)

THE GOOGLE RESEARCH VAN—CONT'D

Broader User Population. We created the van (Figure 10.5) as a response to a key challenge we face doing research on Google premises—user sample. The population in any one geographical area is not always representative of the attitudes and behaviors we'd see in the broader US or international population. We want to reduce as much bias as possible in our product development process, and research with a representative user base is one way to achieve that.

Nimble Recruiting. By driving to locations like malls, community centers, and parks, we can reach users during the natural course of their day without a time-consuming recruitment and scheduling process for us and them. The van presence serves as natural recruitment: individuals who were previously unaware of user studies can walk right up to the van to participate in research. We are also able to reach users who otherwise cannot come to a lab either due to time constraints, accessibility, or other factors, including the many people who are unaware that they can sign up to be contacted for user research. Once we arrive and park our research van, we can invite people passing by, to participate in research studies within minutes.

Nimble Research. Another challenge is the nature of the research itself. In traditional lab research, we recruit and schedule users to participate, sometimes weeks in advance, and ask them to take time out of their day. This process of recruiting and scheduling users can be time consuming; a potentially heavy cost in an industry with quick turnaround and iteration. If participants fail to show up, we lose valuable data and time, and either have to move forward with limited insights or go through the recruiting process again to make up for it.

FIGURE 10.5

The Google user research van.

About the Research Van

The research van is equipped like most other standard usability labs, with an audio and video recording to capture both user reactions and interactions with their mobile device. Considering that research in the field and using a vehicle could add additional stress, the van was designed to be as easy to use as possible. A one button record covers end-to-end and simple default settings allow for easy set up. The cameras capture two key angles: the interactions on the device and the participant's facial and body language. A third video feed includes an HDMI input for the highest resolution view of the device's screen. The researcher can choose to record a full screen of any, or a combination of two in picture-in-picture. A small preview monitor behind the participant allows the moderator to see if the device has gone off camera view and adjusts as necessary, and a large monitor behind the participant allows the note taker to see the video feed clearly for taking notes. The van has enough seats to accommodate the standard moderator, participant, and note taker, as well as two additional stakeholders for product team involvement.

About the Research Tour

We launched the research van with a 6-week cross country research tour in 2016. We went to 10 cities across a range of regions in the US and a variety of locations such as rural towns, college towns, and metropolitan hubs. We were able to meet with people who had never before seen a physical Google presence, much less the opportunity to directly interact with us. In larger cities, we even had the opportunity to talk with tourists from across the country and world. Over 300 people participated in our lab studies inside the van, and we also conducted video interviews and surveys with over 500 participants outside the van. While the research we conducted on our cross country tour was immensely impactful and useful for our product development teams, we can achieve these benefits even closer to home. We can leverage the van by driving to new locations and towns just 1–3 hours outside of Google headquarters, as these regions will also help us reach a much different participant population.

The research van has opened up a world of opportunities for our research practice in reaching a broader user population, increasing flexibility in recruiting, and agility in conducting research. The success of the research van helped us establish new models for recruiting and sourcing as well as brainstorming new standards for our traditional labs. We're excited to see the interest from other areas in Google as well as practitioners industry-wide and we hope to spread the method. We look forward to what more we can do with the research van as a core research infrastructure product and how it can influence research across Google and the industry.

10.5.5 TASK LISTS

Creating the task list can be one of the most challenging parts of creating a usability test. Unless the usability testing is very exploratory, formative, and takes place with very early stage prototypes (possibly on paper), it is likely that a task list will be needed. A task list is used so that when users go through an interface, they are goal-directed. Tasks need to be clear and unambiguous and not need further, additional explanation. While a background scenario may be presented at the beginning of the task list, just to set the participant in the context of the tasks, the task list should not require the participant to ask additional questions about the tasks. The tasks should typically have one clear answer or one clear solution where users know that they have completed the task. Tasks should relate to the key features of the interface, tasks should not be requests for information that the user could know regardless of whether they used the interface (or items that would primarily be found using a web search engine such as Google or Bing). For instance, it would not be appropriate to ask participants to use the interface to find out when Victoria was Queen of England or who won the World Cup in Football in 2012. Participants might already know the answers to these tasks and would not need to use the specific interface. The tasks should clearly require participants to utilize the interface.

Tasks are often chosen based on a number of factors (Dumas and Fox, 2007). For instance, it is important to have tasks that are performed often and are central to the goal that users want to accomplish. In addition, tasks that are critical, such as logging into an account or checking out on an e-commerce site, even if not frequent, should be included. If there are sections of an interface where there are existing questions about usability problems, they could be a focus of some of the tasks. In addition, sometimes, task lists try to be all-inclusive. For instance, if users can utilize menus, shortcuts, or a command line to reach material, some usability moderators design tasks that use all three approaches.

Typically, the task scenarios and the tasks themselves are representative, however they do not utilize any of the user's real data or personal information. Usability testing an interface typically does not involve any of the user's real financial, health, or contact information. Often, test accounts (also known as "dummy" accounts) are created on e-mail servers and transactional servers, so that, as a part of the testing, users will not need to use their own accounts or enter any personal information. These test accounts will be utilized only for the purpose of testing. Even fake identities may be used in usability testing, for instance, when filling out an online form as a part of the usability test; users will be given a fake name, such as "John Smith."

It is important to note a few things about the use of test accounts and fake names. First of all, do not ask users to actually create the fake identities or test accounts, as it will be a waste of time. Have these accounts and fake names already prepared for users. Second, be aware that, while test accounts and fake identities are often utilized in usability testing, there are situations where it could be a violation of law to submit fake information. So, for instance, Wentz et al. noted that, when submitting data to government emergency agencies, even as a part of usability testing of their interfaces, you may not submit fake information, without the express approval of and collaboration with the government agency (Wentz et al., 2014). Third, people with cognitive disabilities may find it confusing to use test accounts or fake names (see Chapter 16 for more information).

Generally, while the moderators may need to have information about how old the users are, their level of education, and their home address (for example, to mail payment for participation), this information is not used as a part of the testing tasks. Furthermore, if a task involves purchasing an item on an e-commerce site, participants should not be required to use their own credit or debit card. Rather, a separate credit card should be provided for their use, so that participants are not charged, and they do not need to provide any personal data. Zazelenchuk et al. (2008) describe how users' real personal data could potentially be used to get a more realistic usability test for financial interfaces, where users are familiar with their own data, emotionally engaged by their own data, and have no trouble understanding the meaning of it. While this may be more realistic from a testing point of view, there are many challenges and logistical concerns to using actual user data, regarding permission to use the data, disposal of the data, and permission from the users themselves. Zazelenchuk et al. even noted that when participants were asked to bring their own financial data to a usability testing session, a number of participants dropped out; to compensate and recruit more people, participants had to be paid a higher amount of money for the extra work and concern. Furthermore, if the usability testing needs to be approved by some sort of institutional review board (see Chapter 15), it is possible that this type of usability testing plan would be rejected. This would be especially likely if the usability testing involved user health care information, as many countries have specific laws relating to the privacy and security of health care information.

When creating the task list, it is important to provide direction on how to navigate the task list itself. Must participants go through tasks in the order listed? Can they skip tasks or start at the last task? Do certain tasks require that other tasks be completed first? How should participants respond if they get stuck and are unable to complete one task? Should they skip that task and move onto the next task? Is there a time limit per task? Is there an overall time limit for the usability testing session? While there might be research reasons for having a time limit per task or per session, there might also be practical reasons. For instance, the supervisor at the workplace may have said that participants can spend no more than 30 minutes on this outside activity or the moderators may only have use of the usability lab space (or another space) for a limited amount of time.

The moderators also need to decide, in advance, whether interventions will be allowed. Interventions are when there is an interface barrier that users are presented with, which does not allow the participant to continue in the interface. The moderator can intervene, if the user gets totally stuck and indicates that they are unable to move on. For instance, if a login screen or an introductory screen is very hard to use, users may not be able to access the rest of the web site or application. If the moderator helps the user move onto the next step, it is still possible to get useful feedback. Interventions specific to people with disabilities and accessibility are described in

Chapter 16. Generally, if researchers do not intervene, this means that the data collection is over, and that would be a missed opportunity to learn more about other aspects of an interface or other aspects of data collection. An intervention is when a researcher helps the participant move forward by providing advice or suggesting a action. Before beginning any usability testing, a researcher should have a clear decision on whether any interventions will be allowed, under what circumstances, how they will be documented, and how this will be accounted for in reporting the results. Typically, the researchers (moderators) don't get involved with providing advice to users, and interventions are not a frequent methodological occurrence. However, the benefit of interventions is that they allow for the maximal amount of feedback about what aspects of the interface need improvement. The details of the intervention should be clearly noted in any data results or write up (Dumas and Fox, 2007)

10.5.6 MEASUREMENT

There are many different types of data that can be collected during usability testing. The three most common quantitative measurements are task performance, time performance, and user satisfaction. Task performance or correctness means how many tasks were correctly completed (and the related metrics of how many tasks were attempted but not successfully completed). Time performance means how long each task took to successfully complete (and the related metrics of how long people spent on incorrect tasks before they gave up). User satisfaction is often measured by a standardized, validated survey tool. See Section 5.8 for a list of standard survey tools for measuring satisfaction.

While these are the three most common quantitative measurements in usability testing, there are many other metrics that could be useful. For instance, additional metrics might include the number of errors, average time to recover from an error, time spent using the help feature, and number of visits to the search feature or index. Depending on the purpose of the usability testing, additional specific metrics might be useful. For instance, if you have redesigned the search engine on a website and the usability testing tasks are focused on the search engine, then an important metric might be something like the average number of search engine responses clicked on, or the average search ranking of the choice that provided the solution. If you utilize key logging, there are many metrics that can be easily analyzed, such as the time spent on specific web pages, the number of web pages viewed, mouse movements, typing speed (Atterer and Schmidt, 2007). See Chapter 12 for information on key logging. Eye tracking used to be prohibitively expensive, but as costs have come down, eye tracking has become more prevalent for usability testing. For more information about eye tracking, see Chapter 13.

In usability testing, especially formative usability testing (on early-stage designs), qualitative data is often just as important as quantitative data. For instance, users are often encouraged to "think aloud" as they are going through the interface (known as the "thinking aloud" protocol). This is more common in formative usability testing than in summative usability testing (when users may be expected to focus more on task completion). When users state their feelings, their frustrations, and their progress out loud, there is often very useful feedback. For instance, a user may say things such as “Where is the menu choice? I would expect it to be right there” or “I certainly would not purchase anything from this website. It looks so unprofessional.” Even younger users can make useful comments during a usability session (see the Leescircus sidebar). It is important to be aware that how comfortable someone may feel about speaking aloud during the tasks may be culturally influenced, and therefore people from some cultures may not feel comfortable expressing their concerns immediately (Shi and Clemmensen, 2008). Also, the more that users talk, the more their task or time performance data may be influenced (Dumas and Loring, 2008). The more they talk, the longer their task times will be (Dumas and Loring, 2008). If you want both true user comments and very accurate task and time performance data, it is possible to run a reflection session, also known as an interpretation session or a retrospective session, after the tasks are performed. In an interpretation session, the users watch raw video of themselves immediately after attempting a series of tasks; working with the evaluators, they interpret the problems they encountered and where they feel that the major interface flaws are (Frokjaer and Hornbæk, 2005). In more traditional research with larger numbers of participants, the goal might be to categorize the qualitative comments using content analysis and look for patterns. With usability testing, we’re trying to use these comments to help improve the interface. Certainly, there is an even more important message for researchers if you hear the same comment multiple times, but the strength of even one comment is important.

USABILITY TESTING OF THE SOFTWARE LEESCIRCUS

Usability testing took place for an educational software package called *Leescircus*, designed for 6- and 7-year-old children in the Netherlands. One example of a typical task was to match pictures that rhyme. A total of 70 Dutch children (32 girls and 38 boys), aged 6 or 7, took part in the usability testing. Most of the children had previous experience with computers and some had previous experience with the program. The children were asked to find problems with this version of the software. There were four sets of eight or nine tasks and each child performed only one set of tasks. Usability evaluators observed the children while they were performing the tasks. The children were encouraged to speak their comments aloud while using the software. The time period was limited to 30 minutes, as it was expected that the attention span of the children wouldn’t last much longer. Although only 28 children did make comments out loud, the novice students (with less computer experience) tended to make more comments than the experts. Usability findings included the need to enlarge the clickable objects, clarify the meaning of icons, and improve consistency (so that it was clear whether an icon could or could not be clicked) (Donker and Reitsma, 2004). This case study shows that children, too, can provide feedback using the “think aloud” protocol during a usability test, although not all will feel comfortable enough to speak up during the usability test

10.5.7 THE USABILITY TESTING SESSION

Before the testing session is scheduled, it is important to contact the participants, remind them about the upcoming session, and confirm the location, regardless of where the usability testing session will take place. Make sure to leave extra time in your schedule, since the participants may show up late, or take longer than expected. Immediately before the session starts, confirm that all computers, recording devices, and other technologies are working properly.

Remember that while the goals of usability testing may be different from classical research like experimental design or ethnography, the protection of human subjects are exactly the same. Just as in any type of research, participants must be given notice of their rights, agree if they are to be video- or audio-recorded, and be allowed to leave at any time. At no point can participants be held against their will, or punished in any way. Unless the participants have specifically given permission to do so, their participation must remain anonymous—at no point can their participation be identified to the outside world. Their data must be protected as in any other type of research method.

It is important to let the participants know if there are any time constraints, either on the session as a whole, or for completing specific tasks. For more information about human subjects protections and IRB forms, see Chapter 15. In usability testing, when new interfaces are being tested, these interfaces might be confidential company information. So participants may also be asked to sign some type of confidentiality agreement, in which they agree not to release or discuss any details of this new interface product (Dumas and Loring, 2008). Finally, it should be clarified before the testing session begins whether participants will receive payment at the end of the session or if a check (or a gift card or something similar) will be mailed to their home. It should also be made clear to the participants that even if they cannot complete the session or feel the need to end the session early, as is common practice, they will still be paid for their participation.

As noted previously, usability testing is about finding flaws that can be fixed, not about having a perfect methodology. One practice that is common in usability testing is to modify the interface after every user test, to help immediately improve the interface flaws discovered; those changes are then evaluated during the next user test (Wixon, 2003). If changes aren't made immediately after each user, changes may be made to the interface after a few users, and then a second round of usability testing is held, using the same tasks, to see if the performance improves and if the changes improved the interface. See the "Usability Testing at Fidelity Investments" sidebar for an example of this practice. Making changes after each user, while an interface is still under development, is commonplace in usability testing. A newer approach to usability testing is A/B testing, where minor tweaks are made in interfaces that are already in daily use. So, for websites that are visited by thousands of users a day, users may receive versions that have slight differences in color, layout, terminology, or other changes that might not be noticeable to the user, with data collected about patterns of usage. After data is collected over perhaps a few weeks, the interface changes that are deemed to be successful, increasing traffic, increasing sales, and reducing costs are permanently rolled out.

USABILITY TESTING AT FIDELITY INVESTMENTS

Usability testing took place at Fidelity Investments, evaluating the prototype of an interface to manage individual benefits, including company benefits, retirement savings, and pensions. A total of 27 participants tested the first prototype (this included both younger and older users, which would be expected for this type of interface). Each participant was given 15 tasks to complete, such as switching retirement money from one plan to another and determining what pension benefits would amount to if the individual retired at 65 years old. Task success rates were 64.2% for users under 55 years old and 44.8% for users aged 55 years or older.

Based on the usability testing of the first prototype, changes were made to the interface, including improved terminology, making links consistently obvious, adding more instruction for detailed table data, adding more titles, and removing false window bottoms and mouseover-based navigation tabs.

Usability testing then took place with the second interface prototype and a new set of 22 participants took part. The new participants had the same profile of age and computer experience as the participants in the first round of testing. The participants in the second round of usability testing were given the same 15 tasks as participants in the first round of testing, with a few minor wording changes due to the updated interface. With the new interface prototype, task success rates improved to 80.6% for users under age 55 and 58.2% for users aged 55 years and older (Chadwick-Dias et al., 2003).

Unlike in other traditional research, in usability testing, it is considered a standard practice to tell participants before they start that they are not being tested. Rather, the participants are testing the interface. Their feedback is important. They are the experts and have the right to criticize the interface. Users are not being tested. You may need to remind them of that fact multiple times.

Note that during the testing session, there are two "tracks" of data collection. One track is the quantitative metrics, such as task and time performance. Moderators may be timing the tasks, data logging may be keeping track, or video recording may be used for later review. The second track is the qualitative data. Sometimes, participants are very talkative and provide a verbal track of what they are doing. If the participants are not very talkative, and if thinking aloud is the methodological goal, moderators should try to encourage them to share more of how they are feeling. However, these reminders should not be often, since the more that the moderator interrupts the user, the more the user feels watched, the more the user's cognitive flow is interrupted, and the more that the user's behavior may deviate from normal. The thinking aloud protocol is more common in formative usability testing than in summative usability testing, since, if quantitative metrics are considered very important by the stakeholders of that interface, the more the participant stops to talk, the more that their task time is interrupted. So while it is acceptable for the users to stop every now and then to describe what they are doing, if the user talks continuously for 10 minutes, clearly, the task performance time is of questionable use.

Since usability testing is a practical approach to solving problems, hybrid approaches are often used. In a reflection or interpretation session, users, immediately after completing a series of tasks, review the raw video with the usability moderators, and help interpret the interface problems (Frokjaer and Hornbæk, 2005). Even without a formal interpretation session, users often make comments about the interface during the debriefing which follows the usability testing session. Without being prompted, users often make comments out loud during the usability testing session. All feedback from users is important data!

In addition, qualitative data, in terms of observation by moderators, is very important. Moderators can often tell a lot about how participants are managing an interface even when the participant is not saying anything. Participants may sigh or grunt and their facial expressions may tell a story of frustration. It is possible to see frustration or anger in the facial expressions of participants. In fact, certain muscle movements in the face are clear signs of stress (Hazlett, 2006). Even without complex interpretation, it is very probable that, if a user keeps moving towards the screen or squinting, the icons or fonts on the screen may be a bit too small.

10.5.8 MAKING SENSE OF THE DATA

Analyzing data from usability testing is similar to analyzing data from any other type of research. However, the goal of the analysis is different. Since usability testing often uses fewer participants, inferential statistics often are not possible; but simple descriptive statistics are possible. With traditional research, the goal is to write up the results in a paper, publish it in a journal, conference proceedings, or book, and help influence future research and design. With usability testing, the goal is often to write up the results and help influence the design of the specific interface that was tested. Sometimes, a presentation about the results is made to a group of developers or managers who have the power to ensure that the interface is changed. The usability testing report (or presentation) should be oriented towards the goal of improving the specific interface and to those who will read it: interface designers, software engineers, project managers, and other managers involved in software development.

The usability test may have uncovered many different interface flaws that should be addressed. However, due to time concerns, not all of these flaws will be improved upon. So while the report should identify all flaws discovered during usability testing, the report should also prioritize which ones are most important to fix. For each flaw identified, the report should describe the problem, present the data from the usability test, identify the priority of the flaw, suggest a fix, and also estimate the time for the fix. Sometimes, data from usability testing can point to which flaws caused users to lose the most time or be unable to complete their tasks and which flaws were easily overcome by users. It is not always clear how to improve every single flaw. Sometimes, you may improve upon one flaw but introduce other problems. An experienced usability moderator may use their expertise to determine which flaws should be prioritized, which flaws are not as problematic, and how to make improvements which do not introduce new problems.

Rubin and Chisnell (2008) suggest splitting the report into three sections:

- why you did usability testing and how you prepared;
- what happened during the testing; and
- the implications and recommendations.

While typical research publications need to be thorough and detailed, if usability testing reports are going to management, they should be short and to the point. If certain aspects of the interface worked well, it might be useful to note that in the report as well. When interface flaws are fixed and changes are made, new flaws can be introduced into the interface. So it can be helpful to note the interface components that worked well and should not be changed.

It is important to note that you should never include names or identifying information for the participants who took part in the usability testing (Dumas and Loring, 2008). If all participants are from within a specific organization, even giving a combination of age, gender, and job title could be the equivalent of identifying someone. When in doubt, provide only the average age of participants, the number of each gender who took part, and basic job titles. You never want to identify who took part in the usability testing, so it's a good idea to refer to the participants as Participant #1, Participant #2, and so on. You never know to whom and where your usability reporting results will be sent to, so make sure you would be comfortable with that fact.

10.6 OTHER VARIATIONS ON USABILITY TESTING

This chapter has presented traditional ways of doing usability testing. But, since usability testing is all about being practical and about changing methods to fit the needs that you have in a project, of course, there are new and different approaches to usability testing. If you read the proceedings of any well-established HCI conference, you can find new approaches, new hybrids, combining multiple methods, that could potentially be used in certain types of usability engineering activities. Two of the more well-known approaches are "technology probes" and "Wizard-of-Oz testing."

Technology probes wouldn't technically be considered usability testing, but they are certainly closer to usability testing than traditional research. A technology probe is similar to a cultural probe (described in Chapter 8). However, a cultural probe has the goal of generally learning more about people, their groups, and their lifestyles. Technology probes involve putting a technology into a real-world setting. Technology probes combine the social science goal of collecting information about people in a real-world setting, the engineering goal of evaluating a new technology, and the design goal of creating new ideas for potential technologies (Hutchinson et al., 2003). A technology is installed in a real-world setting to see how it is used and then reflection on these experiences gives feedback on who the users are and what types of technology could be successfully used in these settings by these users. The technologies themselves are not the interfaces being tested for usability. Technology probes have been used to understand how family members communicate and share images (Hutchinson et al., 2003) and how people in a relationship show public affection (O'Brian and Mueller, 2006). The focus in a technology probe isn't the probe itself but, rather, what can be learned about the people taking part and what technologies they could potentially use.

A Wizard-of-Oz[11] method is essentially a simulation of functionality that doesn't exist yet in an interface application. The user perceives that they are interacting with the actual interface and system. In reality, the user is interacting with another human being that is providing the responses to the user (Dahlback et al., 1993; Gould et al., 1983). Wizard-of-Oz methods can be used when the functionality has not been built due to cost concerns and when the technology doesn't exist, to test potential future interfaces (White and Lutters, 2003). In addition, due to the low time and cost involved, the method may also be helpful in determining feasibility and testing concepts prior to any real systems development (White and Lutters, 2003). Because there can sometimes be a time delay before the "wizard" responds, it can be helpful to have a set of precompiled responses that can quickly be accessed, which helps to improve the realism of the simulation (since participants typically don't know that the functionality isn't being provided by the computer system). The Wizard-of-Oz method has been used in evaluating motion-based computer games for children (Höysniemi et al., 2004), spoken dialog systems in driving vehicle simulators (Hu et al., 2007), and speech recognition systems (Sinha et al., 2001).

10.7 SUMMARY

Usability testing is often known more generally as "user research." Usability testing, typically involves representative users attempting representative tasks in representative environments, on early prototypes or working versions of computer interfaces, with the goal of improving the quality of an interface by finding flaws, areas of the interface that need improvement. In reality, the approaches utilized in usability testing are often the same as those used in classic research. Metrics utilized in usability testing include measurement of task performance and time performance, similar to experimental design, but usability testing often has different end goals. The goal is not to create research that can be generalized to other projects, but rather, to discover specific flaws so that a specific interface can be improved. As an example, making immediate changes to the interface allows for those changes to be evaluated during the next user test, which is considered acceptable in usability testing, but would be considered unacceptable in experimental design. While expert reviews and automated usability testing do help improve interfaces, typically they are not considered HCI research and/or user research, since they do not involve representative

[1] The name comes from the man behind the curtain in the movie *The Wizard of Oz*.

users in the research. Usability testing can involve many different stages of interface development: paper prototypes, wireframes, partially working, or fully functional prototypes. The specific details of the usability testing, such as the stage of prototype development, location of testing, level of formality, task list, number of participants, and the metrics used, will be determined by the budget, timeline, and logistics of the interface development project. The goal is to coordinate closely with the developers of the interface so that the interface problems discovered, will actually translate into changes being made in the interface in a timely manner. Usability testing is focused on practical usage in industry. Professional groups, such as the Usability Experience Professionals Association (www.uxpa.org), provide useful information for practitioners and researchers.

DISCUSSION QUESTIONS

1. Name two ways in which usability testing is similar to experimental design and two ways in which it is different from experimental design.
2. What business factors tend to drive the scope of usability testing?
3. Which should come first, a user-based test or an expert-based test and why?
4. What is a manual check in an automated usability test?
5. What is the difference between a formative usability test and a summative usability test?
6. From a practical point of view, what business factors tend to determine how many participants take part in usability testing?
7. What are the three qualities of a good task in a task list?
8. Why might it be challenging to utilize the user's personal data in a usability test?
9. What are the three most common quantitative measurements in a usability test?
10. What is the "thinking aloud" protocol and is it used more in formative or summative testing?
11. What is a reflection session?
12. What three things do you need to remind participants about before they begin a usability test?
13. Why should you not give any identification information about participants in the final usability testing report?
14. What are two good reasons for using a Wizard-of-Oz approach to testing?
15. How does a technology probe differ from a cultural probe?

RESEARCH DESIGN EXERCISE

Imagine that you are planning a user-based usability test to evaluate a new interface that allows people to track online their medical information, such as blood tests, diagnostics, annual check-ups, and patient visits. Since many governments have set the goal to move to full electronic patient records in the next few years, this is an important project. Doctors will also use this application but, for this exercise, we're focused on patients. Where might you want to recruit potential participants? Would you utilize real patient data in the usability testing? What might five representative tasks be? Since privacy and security of medical data is important, how would you include tasks that assess how comfortable people are with the privacy and security of their data? Where should these usability tests take place? What type of setting would be most authentic and appropriate? How might you compare the usability of this interface with other interfaces for similar tasks? What specific steps might you take to make participants feel more at ease?

REFERENCES

Andreasen, M., Nielsen, H., Schroder, S., Stage, J., 2007. What happened to remote usability testing? An empirical study of three methods. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1405–1414.

Atterer, R., Schmidt, A., 2007. Tracking the interaction of users with AJAX applications for usability testing. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1347–1350.

Als, B., Jensen, J., Skov, M., 2005. Comparison of think-aloud and constructive interaction in usability testing with children. In: Proceedings of the 2005 Conference on Interaction Design and Children, pp. 9–16.

Au, F., Baker, S., Warren, I., Dobbie, G., 2008. Automated usability testing framework. In: Proceedings of the 9th Australasian User Interface Conference, pp. 55–64.

Chadwick-Dias, A., McNulty, M., Tullis, T., 2003. Web usability and age: how design changes can improve performance. In: Proceedings of the ACM Conference on Universal Usability, pp. 30–37.

Dahlback, N., Jonsson, A., Ahrenberg, L., 1993. Wizard of Oz studies: why and how. In: Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI), pp. 193–200.

Donker, A., Reitsma, P., 2004. Usability testing with young children. In: Proceedings of the Interaction Design and Children Conference, pp. 43–48.

Dray, S., Siegel, D., 2004. Remote possibilities?: International usability testing at a distance. Interactions 11 (2), 10–17.

Dumas, J., Fox, J., 2007. Usability testing: current practice and future directions. In: Sears, A., Jacko, J. (Eds.), The Human Computer Interaction Handbook. second ed. Lawrence Erlbaum Associates, New York, pp. 1129–1149.

Dumas, J., Loring, B., 2008. Moderating Usability Tests: Principles and Practices for Interacting. Morgan Kaufmann Publishers, Amsterdam.

Fails, J.A., Guha, M.L., Druin, A., 2012. Methods and techniques for involving children in the design of new technology for children. Foundations and Trends in Human-Computer Interaction 6 (2), 85–166.

Frokjaer, E., Hornbæk, K., 2005. Cooperative usability testing: complementing usability tests with user-supported interpretation sessions. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1383–1386.

Gould, J., Conti, J., Hovanyecz, T., 1983. Composing letters with a simulated listening typewriter. Communications of the ACM 26 (4), 295–308.

Hazlett, R., 2006. Measuring emotional valence during interactive experiences: boys at video game play. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1023–1026.

Hertzum, M., 2016. A usability test is not an interview. Interactions 23 (2), 82–84.

Hollingsed, T., Novick, D., 2007. Usability inspection methods after 15 years of research and practice. In: Proceedings of the ACM Conference on Design of Communication, pp. 249–255.

Hourcade, J.P., 2007. Interaction design and children. Foundations and Trends in Human-Computer Interaction 1 (4), 277–392.

Höysniemi, J., Hämäläinen, P., Turkki, L., 2004. Wizard of Oz prototyping of computer vision based action games for children. In: Proceedings of the 2004 Conference on Interaction Design and Children, pp. 27–34.

Hu, J., Winterboer, A., Nass, C., et al., 2007. Context & usability testing: user-modeled information presentation in easy and difficult driving conditions. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1343–1346.

Hutchinson, H., Mackay, W., Westerlund, B., et al., 2003. Technology probes: inspiring design for and with families. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 17–24.

Hwang, W., Salvendy, G., 2010. Number of people required for usability evaluation: the 10±2 rule. Communications of the ACM 53 (5), 130–133.

Ivory, M., Hearst, M., 2001. The state of the art in automating usability evaluation of user interfaces. ACM Computing Surveys 33 (4), 470–516.

Kuniavsky, M., 2003. Observing the User Experience: A Practitioner's Guide to User Research. Morgan Kaufmann Publishers, San Francisco.

Lazar, J., Williams, V., Gunderson, J., Foltz, T., 2017. Investigating the potential of a dashboard for monitoring U.S. federal website accessibility. In: Proceedings of the 2017 Hawaii International Conference on System Sciences (HICSS), pp. 2428–2437.

Lazar, J., 2006. Web Usability: A User-Centered Design Approach. Addison-Wesley, Boston.

Lewis, J., 2006. Sample sizes for usability tests: mostly math, not magic. Interactions 13 (6), 29–33.

Lindgaard, G., Chattratichart, J., 2007. Usability testing: what have we overlooked? In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1415–1424.

Miller, G., 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63 (2), 81–96.

Nielsen, J., Landauer, T., 1993. A mathematical model of the finding of usability problems. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 206–213.

Nielsen, J., Mack, R. (Eds.), 1994. Usability Inspection Methods. John Wiley & Sons, New York.

O'Brian, S., Mueller, F., 2006. Holding hands over a distance: technology probes in an intimate, mobile context. In: Proceedings of the 2006 OZCHI Conference, pp. 293–296.

Preece, J., Rogers, Y., Sharp, H., 2002. Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons, New York.

Rubin, J., Chisnell, D., 2008. Handbook of Usability Testing, 2nd. ed. Wiley Publishing, Indianapolis.

Sauro, J., Lewis, J.R., 2012. Quantifying the user experience: practical statistics for user research. Elsevier, Amsterdam.

Schmettow, M., 2012. Sample size in usability studies. Communications of the ACM 55 (4), 64–70.

Schusteritsch, R., Wei, C., LaRosa, M., 2007. Towards the perfect infrastructure for usability testing on mobile devices. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 1839–1844.

Shi, Q., Clemmensen, T., 2008. Communication patterns and usability problem finding in cross-cultural thinking aloud usability testing. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 2811–2816.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N., 2017. Designing the user interface: strategies for effective human-computer interaction, sixth ed. Addison-Wesley, Boston, MA.

Sinha, A., Klemmer, S., Chen, J., et al., 2001. SUEDE: iterative, informal prototyping for speech interfaces. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 203–204.

Snyder, C., 2003. Paper prototyping: the fast and easy way to design and refine user interfaces. Morgan Kaufmann Publishers, San Francisco.

Spool, J., Schroeder, W., 2001. Testing web sites: five users is nowhere enough. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 285–286.

Tullis, T., Albert, W., 2008. Measuring the user experience: collecting, analyzing, and presenting usability metrics. Morgan Kaufmann, Amsterdam.

Van Den Haak, M., De Jong, M., Jan Schellens, P., 2003. Retrospective vs. concurrent think-aloud protocols: testing the usability of an online library catalogue. Behaviour & Information Technology 22 (5), 339–351.

Virzi, R., 1992. Refining the test phase of usability evaluation: how many subjects is enough? Human Factors 34 (4), 457–468.

Wentz, B., Lazar, J., Stein, M., Gbenro, O., Holandez, E., Ramsey, A., 2014. Danger, danger! Evaluating the accessibility of web-based emergency alert sign-ups in the northeastern United States. Government Information Quarterly 31 (3), 488–497.

West, R., Lehman, K., 2006. Automated summative usability studies: an empirical evaluation. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 631–639.

White, K., Lutters, W., 2003. Behind the curtain: lessons learned from a Wizard of Oz field experiment. SIGGROUP Bulletin 24 (3), 129–135.

Wixon, D., 2003. Evaluating usability methods: why the current literature fails the practitioner. Interactions 10 (4), 28–34.

Zazelenchuk, T., Sortland, K., Genov, A., Sazegari, S., Keavney, M., 2008. Using participants' real data in usability testing: lessons learned. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 2229–2236.