Usability Testing

TR 29.3820
August 24, 2006

James R. Lewis

IBM Software Group

Boca Raton, Florida

Abstract

Usability testing is an essential skill for usability practitioners – professionals whose primary goal is to provide guidance to product developers for the purpose of improving the ease-of-use of their products. It is by no means the only skill with which usability practitioners must have proficiency, but it is an important one. A recent survey of experienced usability practitioners indicated that usability testing is a very frequently used method, second only to the use of iterative design. One goal of this chapter is to provide an introduction to the practice of usability testing. This includes some discussion of the concept of usability and the history of usability testing, various goals of usability testing, and running usability tests. A second goal is to cover more advanced topics, such as sample size estimation for usability tests, computation of confidence intervals, and the use of standardized usability questionnaires.

ITIRC Keywords

Usability evaluation
Usability testing
Formative
Summative
Sample size estimation
Confidence intervals
Standardized usability questionnaires

NOTE: The contents of this technical report have been published as a chapter in the Handbook of Human Factors and Ergonomics (3rd Edition) – Lewis, J. R. (2006). Usability testing. In G. Salvendy (ed.), Handbook of Human Factors and Ergonomics (pp. 1275-1316). Hoboken, NJ: John Wiley. The most recent version of this technical report is available at http://drjim.0catch.com.

Contents

INTRODUCTION

Usability testing is an essential skill for usability practitioners – professionals whose primary goal is to provide guidance to product developers for the purpose of improving the ease-of-use of their products. It is by no means the *only* skill with which usability practitioners must have proficiency, but it is an important one. A recent survey of experienced usability practitioners (Vredenburg et al., 2002) indicated that usability testing is a very frequently used method, second only to the use of iterative design.

One goal of this chapter is to provide an introduction to the practice of usability testing. This includes some discussion of the concept of usability and the history of usability testing, various goals of usability testing, and running usability tests. A second goal is to cover more advanced topics, such as sample size estimation for usability tests, computation of confidence intervals, and the use of standardized usability questionnaires.

THE BASICS

What is Usability?

The term 'usability' came into general use in the early 1980s. Related terms from that time were 'user friendliness' and 'ease-of-use,' which 'usability' has since displaced in professional and technical writing on the topic (Bevan et al., 1991). The earliest publication (of which I am aware) to include the word 'usability' in its title was Bennett (1979).

It is the nature of language that words come into use with fluid definitions. Ten years after the first use of the term 'usability,' Brian Shackel (1990) wrote, "one of the most important issues is that there is, as yet, no generally agreed definition of usability and its measurement." (p. 31) As recently as 1998, Gray and Salzman stated, "Attempts to derive a clear and crisp definition of usability can be aptly compared to attempts to nail a blob of Jell-O to the wall." (p. 242)

There are several reasons why it has been so difficult to define usability. Usability is not a property of a person or thing. There is no thermometer-like instrument that can provide an absolute measurement of the usability of a product (Dumas, 2003). Usability is an emergent property that depends on the interactions among users, products, tasks and environments.

Introducing a theme that will reappear in several parts of this chapter, there are two major conceptions of usability. These dual conceptions have contributed to the difficulty of achieving a single agreed upon definition. One conception is that the primary focus of usability should be on measurements related to the accomplishment of global task goals (summative, or measurement-based, evaluation). The other conception is that practitioners should focus on the detection and elimination of usability problems (formative, or diagnostic, evaluation).

The first conception has led to a variety of similar definitions of usability, some embodied in current standards (which, to date, have emphasized summative evaluation). For example:

- "The current MUSiC definition of usability is: the ease of use and acceptability of a system or product for a particular class of users carrying out specific tasks in a specific environment; where 'ease of use' affects user performance and satisfaction, and 'acceptability' affects whether or not the product is used." (Bevan et al., 1991, p. 652)
- Usability is the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (ANSI, 2001, p. 3; ISO, 1998, p. 2)
- "To be useful, usability has to be specific. It must refer to particular tasks, particular environments and particular users." (Alty, 1992, p. 105)

One of the earliest formative definitions of usability (ease-of-use) is from Chapanis (1981):

> "Although it is not easy to measure 'ease of use,' it is easy to measure difficulties that people have in using something. Difficulties and errors can be identified, classified, counted, and measured. So my premise is that ease of use is inversely proportional to the number and severity of difficulties people have in using software. There are, of course, other measures that have been used to assess ease of use, but I think the weight of the evidence will support the conclusion that these other dependent measures are correlated with the number and severity of difficulties." (p. 3)

Practitioners in industrial settings generally use both conceptualizations of usability during iterative design. Any iterative method must include a stopping rule to prevent infinite iterations. In the real world, resource constraints and deadlines can dictate the stopping rule (although this rule is valid only if there is a reasonable expectation that undiscovered problems will not lead to drastic consequences). In an ideal setting, the first conception of usability can act as a stopping rule for the second. Setting aside, for now, the question of where quantitative goals come from, the goals associated with the first conception of usability can define when to stop the iterative process of the discovery and resolution of usability problems. This combination is not a new concept. In one of the earliest published descriptions of iterative design, Al-Awar et al. (1981) wrote:

> "Our methodology is strictly empirical. You write a program, test it on the target population, find out what's wrong with it, and revise it. The cycle of test-rewrite is repeated over and over until a satisfactory level of performance is reached. Revisions are based on the performance, that is, the difficulties typical users have in going through the program." (p. 31)

What is Usability Testing?

Imagine the two following scenarios.

Scenario 1: Mr. Smith is sitting next to Mr. Jones, watching him work with a high-fidelity prototype of a web browser for Personal Digital Assistants (PDAs). Mr. Jones is the third person that Mr. Smith has watched performing these tasks with this version of the prototype. Mr. Smith is not constantly reminding Mr. Jones to talk while he works, but is counting on his proximity to Mr. Jones to encourage verbal expressions when Mr. Jones encounters any difficulty in accomplishing his current task. Mr. Smith takes written notes whenever this happens, and also takes notes whenever he observes Mr. Jones faltering in his use of the application (for example, exploring menus in search of a desired function). Later that day he will use his notes to develop problem reports and, in consultation with the development team, will work on recommendations for product changes that should eliminate or reduce the impact of the reported problems. When a new version of the prototype is ready, he will resume testing.

Scenario 2: Dr. White is watching Mr. Adams work with a new version of a word processing application. Mr. Adams is working alone in a test cell that looks almost exactly like an office, except for the large mirror on one wall and the two video cameras overhead. He has access to a telephone and a number to call if he encounters a difficulty that he cannot overcome. If he places such a call, Dr. White will answer and provide help modeled on the types of help provided at the company's call centers. Dr. White can see Mr. Adams through the one-way glass as she coordinates the test. She has one assistant working the video cameras for maximum effectiveness and another who is taking time-stamped notes on a computer (coordinated with the video time stamps) as different members of the team notice and describe different aspects of Mr. Adams' task performance. Software monitors Mr. Adams' computer, recording all keystrokes and mouse movements. Later that day, Dr. White and her associates will put together a summary of the task performance measurements for the tested version of the application, noting where the performance measurements do not meet the test criteria. They will also create a prioritized list of problems and

recommendations, along with video clips that illustrate key problems, for presentation to the development team at their weekly status meeting.

Both of these scenarios provide examples of usability testing. In Scenario 1, the emphasis is completely on usability problem discovery and resolution (formative, or diagnostic evaluation). In Scenario 2, the primary emphasis is on task performance measurement (summative, or measurement-focused evaluation), but there is also an effort to record and present usability problems to the product developers. Dr. White's team knows that they cannot determine if they've met the usability performance goals by examining a list of problems, but they also know that they cannot provide appropriate guidance to product development if they only present a list of global task measurements. The problems observed in the use of an application provide important clues for redesigning the product (Chapanis, 1981; Norman, 1983). Furthermore, as John Karat (1997, p. 693) observed, "The identification of usability problems in a prototype user interface (UI) is not the end goal of any evaluation. The end goal is a redesigned system that meets the usability objectives set for the system such that users are able to achieve their goals and are satisfied with the product."

These scenarios also illustrate the defining properties of a usability test. During a usability test, one or more observers watch one or more participants perform specified tasks with the product in a specified test environment (compare this with the ISO/ANSI definition of usability presented earlier in this chapter). This is what makes usability testing different from other User-Centered Design (UCD) methods. In interviews (including the group interview known as a focus group), participants do not perform work-like tasks. Usability inspection methods (such as expert evaluations and heuristic evaluations), also do not include the observation of users or potential users performing work-like tasks. The same is true of techniques such as surveys and card-sorting. Field studies (including contextual inquiry) can involve the observation of users performing work-related tasks in target environments, but restrict the control that practitioners have over the target tasks and environments. Note that this is not necessarily a bad thing, but it is a defining difference between usability testing and field (ethnographic) studies.

This definition of usability testing permits a wide range of variation in technique (Wildman, 1995). Usability tests can be very informal (as in Scenario 1) or very formal (as in Scenario 2). The observer might sit next to the participant, watch through a one-way glass, or watch the on-screen behavior of a participant who is performing specified tasks at a location halfway around the world. Usability tests can be think-aloud (TA) tests, in which observers train participants to talk about what they're doing at each step of task completion and prompt participants to continue talking if they stop. Observers might watch one participant at a time, or might watch participants work in pairs. Practitioners can apply usability testing to the evaluation of low-fidelity prototypes (see Figure 1), Wizard of Oz (WOZ) prototypes (Kelley, 1985), high-fidelity prototypes, products under development, predecessor products, or competitive products.

Figure 1. Practitioner and participant engaging in an informal usability test with a pencil-and-paper prototype. (Photo courtesy of IBM.)

Where Did Usability Testing Come From?

The roots of usability testing lie firmly in the experimental methods of psychology (in particular, cognitive and applied psychology) and human factors engineering, and are strongly tied to the concept of iterative design. In a traditional experiment, the experimenter draws up a careful plan of study that includes the exact number of participants that the experimenter will expose to the different experimental treatments. The participants are members of the population to which the experimenter wants to generalize the results. The experimenter provides instructions and debriefs the participant, but at no time during a traditional experimental session does the experimenter interact with the participant (unless this interaction is part of the experimental treatment). The more formative (diagnostic, focused on problem discovery) the focus of a usability test, the less it is like a traditional experiment (although the requirements for sampling from a legitimate population of users, tasks, and environments still apply). Conversely, the more summative (focused on measurement) a usability test is, the more it should resemble the mechanics of a traditional experiment. Many of the principles of psychological experimentation that exist to protect experimenters from threats to reliability and validity (for example, the control of demand characteristics) carry over into usability testing (Holleran, 1991; Wenger and Spyridakis, 1989).

As far as I can tell, the earliest accounts of iterative usability testing applied to product design came from Alphonse Chapanis and his students (Al-Awar et al., 1981; Chapanis, 1981; Kelley, 1984), with almost immediate influence on product development practices at IBM (Kennedy, 1982; Lewis, 1982) and other companies, notably Xerox (Smith et al., 1982) and Apple (Williams, 1983). Shortly thereafter, John Gould and his associates at the IBM T. J. Watson Research Center began publishing influential papers on

usability testing and iterative design (Gould, 1988; Gould and Boies, 1983; Gould and Lewis, 1984; Gould et al., 1987).

The driving force that separated iterative usability testing from the standard protocols of experimental psychology was the need to modify early product designs as rapidly as possible (as opposed to the scientific goal of developing and testing competing theoretical hypotheses). As Al-Awar et al. (1981) reported, "Although this procedure [iterative usability test, redesign, and retest] may seem unsystematic and unstructured, our experience has been that there is a surprising amount of consistency in what subjects report. Difficulties are not random or whimsical. They do form patterns." (p. 33)

When, during the early stages of iterative design, difficulties of use become apparent, it is hard to justify continuing to ask test participants to perform the test tasks. There are ethical concerns with intentionally frustrating participants who are using a product with known flaws that the design team can and will correct. There are economic concerns with the time wasted by watching participants who are encountering and recovering from known error-producing situations. Furthermore, any delay in updating the product delays the potential discovery of problems associated with the update or problems whose discovery was blocked by the presence of the known flaws. For these reasons, the earlier you are in the design cycle, the more rapidly you should iterate the cycles of test and design.

Is Usability Testing Effective?

The widespread use of usability testing is evidence that practitioners believe that usability testing is effective. Unfortunately, there are fields in which practitioners' belief in the effectiveness of their methods does not appear to be warranted by those outside of the field (for example, the use of projective techniques such as the Rorschach test in psychotherapy, Lilienfeld et al., 2000). In our own field, a number of recently published papers have questioned the reliability of usability problem discovery (Kessner et al., 2001; Molich et al., 1998, 2004).

The common finding in these studies has been that observers (either individually or in teams across usability laboratories) who evaluated the same product produced markedly different sets of discovered problems. Molich et al. (1998) had four independent usability laboratories carry out inexpensive usability tests of a software application for new users. The four teams reported 141 different problems, with only one problem common among all four teams. Molich et al. (1998) attributed this inconsistency to variability in the approaches taken by the teams (task scenarios, level of problem reporting). Kessner et al. (2001) had six professional usability teams independently test an early prototype of a dialog box. None of the problems were detected by every team, and 18 problems were described by one team only. Molich et al. (2004) assessed the consistency of usability testing across nine independent organizations that evaluated the same website. They documented considerable variability in methodologies, resources applied, and problems reported. The total number of reported problems was 310, with only two problems reported by six or more organizations, and 232 problems uniquely reported. "Our main conclusion is that our simple assumption that we are all doing the same and getting the same results in a usability test is plainly wrong" (Molich et al., 2004, p. 65).

This is important and disturbing research, but there is a clear need for much more research in this area. A particularly important goal of future research should be to reconcile these studies with the documented reality of usability improvement achieved through iterative application of usability testing. For example, a limitation of research that stops with the comparison of problem lists is that it is not possible to assess the magnitude of the usability improvement (if any) that would result from product redesigns based on design recommendations derived from the problem lists (Wixon, 2003). When comparing problem lists from many labs, one aberrant set of results can have an extreme effect on measurements of consistency across labs, and the more labs that are involved, the more likely this is to happen.

The results of these studies (Kessner et al., 2001; Molich et al., 1998, 2004) stand in stark contrast to the published studies in which iterative usability tests (sometimes in combination with other UCD methods) have led to significantly improved products (Al-Awar et al., 1981; Bailey, 1993; Bailey et al., 1992;

Gould et al., 1987; Kelley, 1984; Kennedy, 1982; Lewis, 1982; Lewis, 1996b; Ruthford and Ramey, 2000). For example, in a paper describing their experiences in product development, Marshall et al. (1990) stated, "Human factors work can be reliable – different human factors engineers, using different human factors techniques at different stages of a product's development, identified many of the same potential usability defects" (p. 243). Published cost-benefit analyses (Bias and Mayhew, 1994) have demonstrated the value of usability engineering processes that include usability testing, with cost-benefit ratios ranging from 1:2 for smaller projects to 1:100 for larger projects (C. Karat, 1997).

Most of the papers that describe the success of iterative usability testing are case studies (such as Marshall et al., 1990), but a few have described designed experiments. Bailey et al. (1992) compared two user interfaces derived from the same base interface – one modified via heuristic evaluation and the other modified via iterative usability testing (three iterations, five participants per iteration). They conducted this experiment with two interfaces, one character-based and the other a graphical user interface (GUI), with the same basic outcomes. The number of changes indicated by usability testing was much smaller than the number indicated by heuristic evaluation, but user performance was the same with both final versions of the interface. All designs after the first iteration produced faster performance than and, for the character-based interface, were preferred to, the original design. The time to complete the performance testing was about the same as that required for the completion of multi-reviewer heuristic evaluations.

Bailey (1993) provided additional experimental evidence that iterative design based on usability tests leads to measurable improvements in the usability of an application. In the experiment, he studied the designs of eight designers, four with at least four years of professional experience in interface design and four with at least five years of professional experience in computer programming. All designers used a prototyping tool to create a recipes application (eight applications in all). In the first wave of testing, Bailey videotaped participants performing tasks with the prototypes, three different participants per prototype. Each designer reviewed the videotapes of the people using his or her prototype, and used the observations to redesign his or her application. This process continued until each designer indicated that it was not possible to improve his or her application. All designers stopped after three to five iterations. Comparison of the first and last iterations indicated significant improvement in measurements such as number of tasks completed, task completion times, and repeated serious errors.

In conclusion, the results of the studies of Molich et al. (1998, 2004) and similar studies show that usability practitioners must conduct their usability tests as carefully as possible, document their methods completely, and show proper caution when interpreting their results. On the other hand, as Landauer stated in 1997, "There is ample evidence that expanded task analysis and formative evaluation can, and almost always do, bring substantial improvements in the effectiveness and desirability of systems" (p. 204). This is echoed by Desurvire et al. (1992, p. 98), "It is generally agreed that usability testing in both field and laboratory, is far and above the best method for acquiring data on usability."

Goals of Usability Testing

The fundamental goal of usability testing is to help developers produce more usable products. The two conceptions of usability testing (formative and summative) lead to differences in the specification of goals in much the same way that they contribute to differences in fundamental definitions of usability (diagnostic problem discovery and measurement). Rubin (1994, p. 26) expressed the formative goal as, "The overall goal of usability testing is to identify and rectify usability deficiencies existing in computer-based and electronic equipment and their accompanying support materials prior to release." Dumas and Redish (1999, p. 11) provided a more summative goal with, "A key component of usability engineering is setting specific, quantitative, usability goals for the product early in the process and then designing to meet those goals."

These goals are not in direct conflict, but they do suggest different foci that can lead to differences in practice. For example, a focus on measurement typically leads to more formal testing (less interaction between observers and participants) whereas a focus on problem discovery typically leads to less formal testing (more interaction between observers and participants). In addition to the distinction between diagnostic problem discovery and measurement tests, there are two common types of measurement tests: (1) comparison against objectives and (2) comparison of products.

Problem Discovery Test

The primary activity in diagnostic problem discovery tests is the discovery, prioritization, and resolution of usability problems. The number of participants in each iteration of testing should be fairly small, but the overall test plan should be for multiple iterations, each with some variation in participants and tasks. When the focus is on problem discovery and resolution, the assumption is that more global measures of user performance and satisfaction will take care of themselves (Chapanis, 1981). The measurements associated with problem-discovery tests are focused on prioritizing problems, and include frequency of occurrence in the test, likelihood of occurrence during normal usage (taking into account the anticipated usage of the part of the product in which the problem occurred), and magnitude of impact on the participants who experienced the problem. Because the focus is not on precise measurement of the performance or attitudes of participants, problem discovery studies tend to be informal, with a considerable amount of interaction between observers and participants. Some typical stopping rules for iterations are a preplanned number of iterations or a specific problem discovery goal, such as "Identify 90% of the problems available for discovery for these types of participants, this set of tasks, and these conditions of use." See the section below on sample size estimation and adequacy for more detailed information on setting and using these types of problem-discovery objectives.

Measurement Test Type I: Comparison against Quantitative Objectives

Studies that have a primary focus of comparison against quantitative objectives include two fundamental activities. The first is the development of the usability objectives. The second is iterative testing to determine if the product under test has met the objectives. A third activity (which can take place during iterative testing) is the enumeration and description of usability problems, but this activity is secondary to the collection of precise measurements.

The first step in developing quantitative usability objectives is to determine the appropriate variables to measure. Rengger (1991), as part of the work done for the European MUSiC project (Measuring the Usability of Systems in Context) produced a list of potential usability measurements based on 87 papers out of a survey of 500 papers. He excluded purely diagnostic studies and also excluded papers if they did not provide measurements for the combined performance of a user and a system. He categorized the measurements into four classes:

- Class 1: Goal achievement indicators (such as success rate and accuracy)
- Class 2: Work rate indicators (such as speed and efficiency)
- Class 3: Operability indicators (such as error rate and function usage)
- Class 4: Knowledge acquisition indicators (such as learnability and learning rate)

In a later discussion of the MUSiC measures, Macleod et al. (1997) described measures of effectiveness (the level of correctness and completeness of goal achievement in context) and efficiency (effectiveness related to cost of performance – typically the effectiveness measure divided by task completion time). Optional measures were of productive time and unproductive time, with unproductive time consisting of help actions, search actions, and snag (negation, cancelled, or rejected) actions.

Their (Macleod et al., 1997) description of the measures of effectiveness and efficiency seem to have influenced the objectives expressed in ISO 9241-11 (1998, p. iv): "The objective of designing and evaluating visual display terminals for usability is to enable users to achieve goals and meet needs in a particular context of use. ISO 9241-11 explains the benefits of measuring usability in terms of user performance and satisfaction. These are measured by the extent to which the intended goals of use are

achieved, the resources that have to be expended to achieve the intended goals, and the extent to which the user finds the use of the product acceptable."

In practice (and as recommended in the ANSI Common Industry Format for Usability Test Reports, 2001), the fundamental global measurements for usability tasks are successful task completion rates (for a measure of effectiveness), mean task completion times (for a measure of efficiency), and mean participant satisfaction ratings (either collected on a task-by-task basis or at the end of a test session – see the section below on standardized usability questionnaires for more information on measuring participant satisfaction). There are many other measurements that practitioners could consider (Dumas and Redish, 1999; Nielsen, 1997), including but not limited to:

1. The number of tasks completed within a specified time limit
2. The number of wrong menu choices
3. The number of user errors
4. The number of repeated errors (same user committing the same error more than once)

After determining the appropriate measurements, the next step is to set the goals. Ideally, the goals should have an objective basis and shared acceptance across the various stakeholders, such as Marketing, Development, and Test groups (Lewis, 1982). The best objective basis for measurement goals are data from previous usability studies of predecessor or competitive products. For maximum generalizability, the historical data should come from studies of similar types of participants completing the same tasks under the same conditions (Chapanis, 1988). If this information is not available, then an alternative is for the test designer to recommend objective goals and to negotiate with the other stakeholders to arrive at a set of shared goals.

"Defining usability objectives (and standards) isn't easy, especially when you're beginning a usability program. However, you're not restricted to the first objective you set. The important thing is to establish some specific objectives immediately, so that you can measure improvement. If the objectives turn out to be unrealistic or inappropriate, you can revise them." (Rosenbaum, 1989, p. 211) Such revisions, however, should take place only in the early stages of gaining experience and taking initial measurements with a product. It is important not to change reasonable goals to accommodate an unusable product.

When setting usability goals, it's usually better to set goals that make reference to an average (mean) of a measurement than to a percentile. For example, set an objective such as "The mean time to complete Task 1 will be less than five minutes" rather than "95% of participants will complete Task 1 in less than ten minutes". The statistical reason for this is that sample means drawn from a continuous distribution are less variable than sample medians (the 50^{th} percentile of a sample), and measurements made away from the center of a distribution (for example, measurements made to attempt to characterize the value of the 95^{th} percentile) are even more variable (Blalock, 1972). Cordes (1993) conducted a Monte Carlo study comparing means and medians as measurements of central tendency for time-on-task scores, and determined that the mean should be the preferred metric for usability studies (unless there is missing data due to participants failing to complete tasks, in which case the mean from the study will underestimate the population mean).

A practical reason to avoid percentile goals is that the goal can imply a sample size requirement that is unnecessarily large. For example, you can't measure accurately at the 95^{th} percentile unless there are at least twenty measurements (in fact, there must be many more than twenty measurements for accurate measurement). For more details, see the section below on sample size estimation for measurement.

An exception to this is the specification of successful task completions (or any other measurement that is based on counting events), which necessarily requires a percentile goal, usually set at or near 100% (unless there are historical data that indicate an acceptable lower level for a specific test). If ten out of ten participants complete a task successfully, the observed completion rate is 100%, but a 90% binomial confidence interval for this result ranges from 74% to 100%. In other words, even perfect performance for ten participants with this type of measure leaves open the possibility (with 90% confidence) that the true

completion rate could be as low as 75%. See the section below on binomial confidence intervals for more information on computing and using this information in usability tests.

After the usability goals have been established, the next step is to collect data to determine if the product has met its goals. Representative participants perform the target tasks in the specified environment as test observers record the target measurements and identify, to the extent possible within the constraints of a more formal testing protocol, details about any usability problems that occur. The usability team conducting the test provides information about goal achievement and prioritized problems to the development team, and a decision is made regarding whether or not there is sufficient evidence that the product has met its objectives. The ideal stopping rule for measurement-based iterations is to continue testing until the product has met its goals.

When there are only a few goals, then it is reasonable to expect to achieve all of them. When there are many goals (for example, five objectives per task multiplied by ten tasks for a total of fifty objectives), then it is more difficult to determine when to declare success and to stop testing. Thus, it is sometimes necessary to specify a meta-objective of the percentage of goals to achieve.

Despite the reluctance of some usability practitioners to conduct statistical tests to quantitatively assess the strength of the available evidence regarding whether or not a product has achieved a particular goal, the best practice is to conduct such tests. The best approach is to conduct multiple *t*-tests or nonparametric analogs of *t*-tests (Lewis, 1993) because this gives practitioners the level of detail that they require. There is a well-known prohibition against doing this because it can lead investigators to mistakenly accept as real that some differences that are due to chance (technically, alpha inflation). On the other hand, if this is the required level of information, then it is an appropriate method (Abelson, 1995). Furthermore, the practice of avoiding alpha inflation is a concern more related to scientific hypothesis testing than to usability testing (Wickens, 1998), although usability practitioners should be aware of its existence and take it into account when interpreting their statistical results. For example, if you compare two products by conducting fifty *t*-tests with alpha set to .10, and only five (10%) of the *t*-tests are significant (have *p* less than .10), then you should question whether or not to use those results as evidence of the superiority of one product over the other. On the other hand, if substantially more than five of the *t*-tests are significant, then you can be more confident that the indicated differences are real.

In addition to (or as an alternative to) conducting multiple *t*-tests, practitioners should compute confidence intervals for their measurements. This applies to the measurements made for the purpose of establishing test criteria (such as measurements made on predecessor versions of the target product or competitive products) and to the measurements made when testing the product under development. See the section below on confidence intervals for more details.

Measurement Test Type II: Comparison of Products

The second type of measurement test is to conduct usability tests for the purpose of directly comparing one product with another. As long as there is only one measurement that decision makers plan to consider, then a standard *t*-test (ideally, in combination with the computation of confidence intervals) will suffice for the purpose of determining which product is superior.

If decision makers care about multiple dependent measures, then standard multivariate statistical procedures (such as MANOVA or discriminant analysis) are not often helpful in guiding a decision about which of two products has superior usability. The statistical reason for this is that multivariate statistical procedures depend on the computation of centroids (a weighted average of multiple dependent measures) using a least-squares linear model that maximizes the difference between the centroids of the two products (Cliff, 1987). If the directions of the measurements are inconsistent (for example, a high task completion rate is desirable, but a high mean task completion time is not), then the resulting centroids are uninterpretable for the purpose of usability comparison. In some cases it is possible to recompute variables so they have consistent directions (for example, recomputing task completion rates as task failure rates). If this is not possible, then another approach is to convert measurements to ranks (Lewis, 1991a) or standardized (*Z*)

scores (Jeff Sauro, personal communication, March 1, 2004) for the purpose of principled combination of different types of measurements.

To help consumers compare the usability of different products, the American National Standards Institute (ANSI) has published the Common Industry Format (CIF) for usability test reports (ANSI, 2001). Originally developed at the National Institute of Standards and Technology (NIST), this test format requires measurement of effectiveness (accuracy and completeness – completion rates, errors, assists), efficiency (resources expended in relation to accuracy and completeness – task completion time), and satisfaction (freedom from discomfort, positive attitude toward use of the product – using any of a number of standardized satisfaction questionnaires). It also requires a complete description of participants and tasks.

Morse (2000) reviewed the NIST IUSR project conducted to pilot test the CIF. The purpose of the CIF is to make it easier for purchasers to compare the usability of different products. The pilot study ran into problems, such as inability to find a suitable software product for both supplier and consumer, reluctance to share information, and uncertainty about how to design a good usability study. To date, there has been little if any use (at least, no published use) of the CIF for its intended purpose.

Variations on a Theme: Other Types of Usability Tests

Think Aloud

In a standard, formal usability test, test participants perform tasks without necessarily speaking as they work. The defining characteristic of a Think Aloud (TA) study is the instruction to participants to talk about what they are doing as they do it (in other words, to produce verbal reports). If participants stop talking (as commonly happens when they become very engaged in a task), they are prompted to resume talking.

The most common theoretical justification for the use of TA is from the work in cognitive psychology (specifically, human problem solving) of Ericsson and Simon (1980). Responding to a review by Nisbett and Wilson (1977) that described various ways in which verbal reports were unreliable, Ericsson and Simon provided evidence that certain kinds of verbal reports could produce reliable data. They stated that reliable verbalizations are those that participants produce during task performance that do not require additional cognitive processing beyond the processing required for task performance and verbalization.

Some discussions of usability testing hold that the best practice in usability testing is to use the TA method in all usability testing. For example, Dumas (2003) encouraged the use of TA because (1) TA tests are more productive for finding usability problems (Virzi, Sorce, and Herbert, 1993) and (2) thinking aloud does not affect user ratings or performance (Bowers and Snyder, 1990). As the references indicate, there is some evidence in support of these statements, but the evidence is mixed.

Earlier prohibitions against the use of TA in measurement-based tests assumed that thinking aloud would cause slower task performance. Bowers and Snyder (1990), however, found no measurable task performance or preference differences between a test group that thought aloud and one that didn't. Surprisingly, there are some experiments in which the investigators reported better task performance when participants were thinking aloud. Berry and Broadbent (1990) provided evidence that the process of thinking aloud invoked cognitive processes that improved rather than degraded performance, but only if people were given (1) verbal instructions on how to perform the task and (2) the requirement to justify each action aloud. Wright and Converse (1992) compared silent with TA usability testing protocols. The results indicated that the think-aloud group committed fewer errors and completed tasks faster than the silent group, and the difference between the groups increased as a function of task difficulty.

Regarding the theoretical justification for and typical practice of TA, Boren and Ramey (2000) noted that TA practice in usability testing often does not conform to the theoretical basis most often cited for it (Ericsson and Simon, 1980). "If practitioners do not uniformly apply the same techniques in conducing thinking-aloud protocols, it becomes difficult to compare results between studies." (Boren and Ramey, 2000,

p. 261) In a review of publications of TA tests and field observations of practitioners running TA tests, they reported inconsistency in explanations to participants about how to think aloud, practice periods, styles of reminding participants to think aloud, prompting intervals, and styles of intervention. They suggest that rather than basing current practice on Ericsson and Simon, a better basis would be speech communication theory, with clearly defined communicative roles for the participant (in the role of domain expert or valued customer, making the participant the primary speaker) and the usability practitioner (the learner or listener, thus, a secondary speaker).

Based on this alternative perspective for the justification of TA, Boren and Ramey (2000) have provided guidance for many situations that are not relevant in a cognitive psychology experiment, but are in usability tests. For example, they recommend that usability practitioners running a TA test should continually use acknowledgement tokens that do not take speakership away from the participant, such as "mm hm?" and "uh-huh?" (with the interrogative intonation) to encourage the participant to keep talking. In normal communication, silence (as recommended by the Ericsson and Simon protocols) is not a nonresponse – the speaker interprets it in a primarily negative way as indicating aloofness or condescension. They avoided providing precise statements about how frequently to provide acknowledgments or somewhat more explicit reminders (such as "And now…?") because the best cues come from the participants. Practitioners need to be sensitive to these cues as they run the test.

The evidence indicates that, relative to silent participation, TA can affect task performance. If the primary purpose of the test is problem discovery, then TA appears to have advantages over completely silent task completion. If the primary purpose of the test is task performance measurement, then the use of TA is somewhat more complicated. As long as all the tasks in the planned comparisons were completed under the same conditions, then performance comparisons should be legitimate. The use of TA almost certainly prevents generalization of task performance outside of the TA task, but there are many other factors that make it difficult to generalize specific task performance data collected in usability studies.

For example, Cordes (2001) demonstrated that participants assume that the tasks they are asked to perform in usability tests are possible (the "I know it can be done or you wouldn't have asked me to do it" bias). Manipulations that bring this assumption into doubt can have a strong effect on quantitative usability performance measures, such as increasing the percentage of participants who give up on a task. If uncontrolled, this bias makes performance measures from usability studies unlikely to be representative of real-world performance when users are uncertain as to whether the product they are using can support the desired tasks.

Multiple Simultaneous Participants

Another way to encourage participants to talk during task completion is to have them work together (Wildman, 1995). This strategy is similar to TA in its strengths and limitations.

Hackman and Biers (1992) compared three think-aloud methods: thinking aloud alone (Single), thinking aloud in the presence of an observer (Observer), and verbalizations occurring in a two-person team (Team). They found no significant differences in performance or subjective measures. The Team condition produced more statements of value to designers than the other two conditions, but this was probably due to the differing number of participants producing statements in the different conditions. There were three groups, with 10 participants per group for Single and Observer, and 20 participants (10 two-person teams) for the Team condition. "The major result was that the team gave significantly more verbalizations of high value to designers and spent more time making high value comments. Although this can be reduced to the fact that the team spoke more overall and that there are two people talking rather than one, this finding is not trivial." (p. 1208)

Remote Evaluation

Recent advances in the technology of collaborative software have made it easier to conduct remote software tests (tests in which the usability practitioner and the test participant are in different locations). This can be an economical alternative to bringing one or more users into a lab for face-to-face user testing.

A participant in a remote location can view the contents of the practitioner's screen, and in a typical system the practitioner can decide whether the participant can control the desktop. System performance is typically slower than that of a local test session.

Some of the advantages of remote testing are (1) access to participants who would otherwise be unable to participate (international, special needs, etc.), (2) the capability for participants to work in familiar surroundings, and (3) no need for either party to install or download additional software. Some of the disadvantages are (1) potential uncontrolled disruptions in the participant's workplace, (2) lack of visual feedback from the participant, and (3) the possibility of compromised security if the participant takes screen captures of confidential material. Despite these disadvantages, McFadden et al. (2002) reported data that indicated that remote testing was effective at improving product designs and that the test results were comparable to the results obtained with more traditional testing.

Usability Laboratories

A typical usability laboratory test suite is a set of soundproofed rooms with a participant area and observer area separated by a one-way glass, and with video cameras and microphones to capture the user experience (Marshall et al., 1990; Nielsen, 1997), possibly with an executive viewing area behind the primary observers' area. The advantages of this type of usability facility are quick setup, a place where designers can see people interacting with their products, videos to provide a historical record and backup for observers, and a professional appearance that raises awareness of usability and reassures customers about commitment to usability. In a survey of usability laboratories, Nielsen (1994) reported a median floor space of 63 m^2 (678 ft^2) for the observer room and 13 m^2 (144 ft^2) for test rooms. This type of laboratory (see Figure 2) is especially important if practitioners plan to conduct formal, summative usability tests.

Figure 2. View of a usability laboratory. (Photo courtesy of IBM.)

If the practitioner focus is on formative, diagnostic problem discovery, then this type of laboratory is not essential (although still convenient). "It is possible to convert a regular office temporarily into a usability laboratory, and it is possible to perform usability testing with no more equipment than a notepad." (Nielsen, 1997, p. 1561) Making an even stronger statement against the perceived requirement for formal laboratories, Landauer (1997, p. 204) stated, "Many usability practitioners have demanded greater resources and more elaborate procedures than are strictly needed for effective guidance – such as expensive usability labs rather than natural settings for test and observations, time consuming videotaping and analysis where observation and note-taking would serve as well, and large groups of participants to achieve statistical significance when qualitative naturalistic observation of task goals and situations, or of disastrous interface or functionality flaws, would be more to the point."

Test Roles

There are several ways to categorize the roles that testers need to play in the preparation and execution of a usability test (Dumas and Redish, 1999; Rubin, 1994). Most test teams will not have an individual assigned to each role, and most tests (especially informal problem discovery tests) do not require every role. The actual distribution of skills across a team might vary from these roles, but the standard roles help to organize the skills needed for effective usability testing.

Test Administrator

The test administrator is the usability test team leader. He or she designs the usability study, including the specification of the initial conditions for a test session and the codes to use for data logging. The test administrator's duties include conducting reviews with the rest of the test team, leading in the analysis of data, and putting together the final presentation or report. People in this role should have a solid understanding of the basics of usability engineering, ability to tolerate ambiguity, flexibility (knowing when to deviate from the plan), and good communication skills.

Briefer

The briefer is the person who interacts with the participants (briefing them at the start of the test, communicating with them as required during the test, and debriefing them at the end of the test sessions). On many teams, the same person takes the roles of administrator and briefer. In a think-aloud study, the briefer has the responsibility to keep the participant talking. The briefer needs to have sufficient familiarity with the product to be able to decide what to tell participants when they ask questions. People in this role need to be comfortable interacting with people, and need to be able to restrict their interactions to those that are consistent with the purposes of the test without any negative treatment of the participants.

Camera Operator

The camera operator is responsible for running the audio-visual equipment during the test. He or she must be skilled in the setup and operation of the equipment, and must be able to take directions quickly when it is necessary to change the focus of the camera (for example, from the keyboard to the user manual).

Data Recorder

The video record is useful as a data backup when things start happening quickly during the test, and as a source for video examples when documenting usability problems. The primary data source for a usability study, however, is the notes that the data recorder takes during a test session. There just isn't time to take notes from a more leisurely examination of the video record. Also, the camera doesn't necessarily catch the important action at every moment of a usability study.

For informal studies, the equipment used to record data might be nothing more than a notepad and pencil. Alternatively, the data recorder might use data-logging software to take coded notes (often time-stamped, possibly synchronized with the video). Before the test begins, the data recorder needs to prepare the data-logging software with the category codes defined by the test administrator. Taking notes with data-logging software is a very demanding skill, so the test administrator does not usually assign additional tasks to the person taking this role.

Help Desk Operator

The help desk operator takes calls from the participant if the user experiences enough difficulty to place the call. The operator should have some familiarity with the call-center procedures followed by the company that has designed the product under test, and must also have skills similar to those of the briefer.

Product Expert

The product expert maintains the product and offers technical guidance during the test. The product expert must have sufficient knowledge of the product to recover quickly from product failures and to help the other team members understand the system's actions during the test.

Statistician

A statistician has expertise in measurement and the statistical analysis of data. Practitioners with an educational background in experimental psychology typically have sufficient expertise to take the role of statistician for a usability test team. Informal tests rarely require the services of a statistician, but the team needs a statistician to extract the maximum amount of information from the data gathered during a formal test (especially if the purpose of the formal test was to compare two products using a battery of measurements).

Planning the Test

One of the first activities a test administrator must undertake is to develop a test plan. To do this, the administrator must understand the purpose of the product, the parts of the product that are ready for test, the types of people who will use the product, what they are likely to use the product for, and in what settings.

Purpose of Test

At the highest level, is the primary purpose of the test to identify usability problems or to gather usability measurements? The answer to this question provides guidance as to whether the most appropriate test is formal or informal, think-aloud or silent, problem discovery or quantitative measurement. After addressing this question, the next task is to define any more specific test objectives. For example, an objective for an interactive voice response system (IVR) might be to assess whether participants can accomplish key tasks without encountering significant problems. If data are available from a previous study of a similar IVR, an alternative objective might be to determine whether participants can complete key tasks reliably faster with the new IVR than they did with the previous IVR. Most usability tests will include several objectives.

If a key objective of the test is to compare two products, then an important decision is whether the test will be within-subjects or between-subjects. In a within-subjects test, every participant works with both products, with half of the participants using one product first and the other half using the other product first (a technique known as counterbalancing). In a between-subjects study, the test groups are completely independent. In general, a within-subjects test leads to more precise measurement of product differences (requiring a smaller number of participants for equal precision, primarily due to the reduction in variability that occurs when each participant acts as his or her own control) and the opportunity to get direct subjective product comparisons from participants. For a within-subjects test to be feasible, both products must be available and set up for use in the lab at the same time, and the amount of time needed to complete tasks with both products must not be excessive. If a within-subjects test is not possible, a between-subjects test is a perfectly valid alternative. Note that the statistical analyses appropriate for these two types of tests are different.

Participants

To determine who will participate in the test, the administrator needs to obtain or develop a user profile. A user profile is sometimes available from the marketing group, the product's functional specification, or other product planning documentation. It is important to keep in mind that the focus of a usability test is the end user of a product, not the expected product purchaser (unless the product will be purchased by end users). The most important participant characteristic is that the participant is representative of the population of end-users to whom the administrator wants to generalize the results of the test. Practitioners can obtain participants from employment agencies, internal sources if the participants meet the requirements of the user profile (but avoiding internal test groups), market research firms, existing customers, colleges, newspaper ads, and user groups.

To define representativeness, it is important to specify the characteristics that members of the target population share but are not characteristic of nonmembers. The administrator must do this for the target population at large and any defined subgroups. Within group definition constraints, administrators should seek heterogeneity in the final sample to maximize the generalizability of the results (Chapanis, 1988; Landauer, 1997) and to maximize the likelihood of problem discovery. It is true that performance measurements made with a homogeneous sample will almost always have greater precision than measurements made with a heterogeneous sample, but the cost of that increased precision is limited generalizability. This raises the issue of how to define homogeneity and heterogeneity of participants. After all, at the highest level of categorization, we are all humans, with similar general capabilities and limitations (physical and cognitive). At the other end of the spectrum, we are all individuals – no two alike.

One of the most important defining characteristics for a group in a usability test is specific relevant experience, both with the product and in the domain of interest (work experience, general product experience, specific product experience, experience with the product under test, and experience with similar products). One common categorization scheme is to consider people with less than three months experience as novices, with more than a year of experience as expert, and those in between as intermediate (Dumas and Redish, 1999). Other individual differences that practitioners routinely track and attempt to vary are education level, age, and sex.

When acquiring participants, how can practitioners define the similarity between the participants they can acquire and the target population? An initial step is to develop a taxonomy of the variables that affect human performance (where performance should include the behaviors of indicating preference and other choice behaviors). Gawron et al. (1989) produced a human performance taxonomy during the development of a human performance expert system. They reviewed existing taxonomies and filled in some missing pieces. They structured the taxonomy as having three top levels: environment, subject (person), and task. The resulting taxonomy took up 12 pages in their paper, and covered many areas which would normally not concern a usability practitioner working in the field of computer system usability (for example, ambient vapor pressure, gravity, acceleration, etc.). Some of the key human variables in the Gawron et al. (1989) taxonomy that could affect human performance with computer systems are:

- Physical Characteristics
  - Age
  - Agility
  - Handedness
  - Voice
  - Fatigue
  - Gender
  - Body and body part size
- Mental State
  - Attention span
  - Use of drugs (both prescription and illicit)
  - Long-term memory (includes previous experience)
  - Short-term memory
  - Personality traits
  - Work schedule
- Senses
  - Auditory acuity
  - Tone perception
  - Tactual
  - Visual accommodation
  - Visual acuity
  - Color perception

These variables can guide practitioners as they attempt to describe how participants and target populations are similar or different. The Gawron et al. (1989) taxonomy does not provide much detail with regard to some individual differences that other researchers have hypothesized to affect human performance or preference with respect to the use of computer systems: personality traits and computer-specific experience.

Aykin and Aykin (1991) performed a comprehensive review of the published studies to that date that involved individual differences in human-computer interaction (HCI). Table 1 lists the individual differences that they found in published HCI studies, the method used to measure the individual difference, and whether there was any indication from the literature that manipulation of that individual difference led to a crossed interaction.

In statistical terminology, an interaction occurs whenever an experimental treatment has a different magnitude of effect depending on the level of a different, independent experimental treatment. A crossed interaction occurs when the magnitudes have different signs, indicating reversed directions of effects. As an example of an uncrossed interaction, consider the effect of turning off the lights on the typing throughput of blind and sighted typists. The performance of the sighted typists would probably be worse, but the presence or absence of light shouldn't affect the performance of the blind typists. As an extreme example of a crossed interaction, consider the effect of language on task completion for people fluent only in French or English. When reading French text, French speakers would outperform English speakers, and vice versa.

Table 1. Results of Aykin and Aykin (1991) review of individual differences in HCI

Individual Difference	**Measurement Method**	**Crossed Interactions**
Level of experience	Various methods	No
Jungian personality types	Myers-Briggs Type Indicator	No
Field dependence/ independence	Embedded Figures Test	Yes – field dependent participants preferred organized sequential item number search mode, but field independent subjects preferred the less organized keyword search mode (Fowler et al., 1985)
Locus of control	Levenson test	No
Imagery	Individual Differences Questionnaire	No
Spatial ability	VZ-2	No
Type A/Type B personality	Jenkins Activity Survey	No
Ambiguity tolerance	Ambiguity Tolerance Scale	No
Sex	Unspecified	No
Age	Unspecified	No
Other (reading speed and comprehension, intelligence, mathematical ability)	Unspecified	No

For any of these individual differences, the lack of evidence for crossed interactions could be due to a paucity of research involving the individual difference or could reflect the probability that individual differences will not typically cause crossed interactions in HCI. In general, a change made to support a problem experienced by a person with a particular individual difference will either help other users or simply not affect their performance.

For example, John Black (personal communication, 1988) cited the difficulty that field dependent users had working with one-line editors at the time (decades ago) when that was the typical user interface to a mainframe computer. Switching to full-screen editing resulted in a performance improvement for both field dependent and independent users – an uncrossed interaction because both types of users improved, with the performance of field dependent users becoming equal to (thus improving more than) that of field independent users. Landauer (1997) cites another example of this, in which Greene et al. (1986) found that young people with high scores on logical reasoning tests could master database query languages such as SQL with little training, but older or less able people could hardly ever master these languages. They also determined that an alternative way of forming queries, selecting rows from a truth table, allowed almost everyone to make correct specification of queries, independent of their abilities. Because this redesign improved the performance of less able users without diminishing the performance of the more able, it was an uncrossed interaction. In a more recent study, Palmquist and Kim (2000) found that field dependence

affected the search performance of novices using a web browser (with field independent users searching more efficiently), but did not affect the performance of more experienced users.

If there is a reason to suspect that an individual difference will lead to a crossed interaction as a function of interface design, then it could make sense to invest the time (which can be considerable) to categorize users according to these dimensions. Another situation in which it could make sense to invest the time in categorization by individual difference would be if there were reasons to believe that a change in interface would greatly help one or more groups without adversely affecting other groups. (This is a strategy that one can employ when developing hypotheses about ways to improve user interfaces.) It always makes sense to keep track of user characteristics when categorization is easy (for example, age or sex). Another potential use of these types of variables is as covariates (used to reduce estimates of variability) in advanced statistical analyses (Cliff, 1987).

Aykin and Aykin (1991) reported effects of users' levels of experience, but did not report any crossed interactions related to this individual difference. They did report that interface differences tended to affect the performance of novices, but had little effect on the performance of experts. It appears that behavioral differences related to user interfaces (Aykin and Aykin, 1991) and cognitive style (Palmquist and Kim, 2000) tend to fade with practice. Nonetheless, user experience has been one of the few individual differences to receive considerable attention in HCI research (Fisher, 1991; Mayer, 1997; Miller et al., 1997; Smith et al., 1999).

According to Mayer (1997), relative to novices, experts have:

- better knowledge of syntax
- an integrated conceptual model of the system
- more categories for more types of routines
- higher level plans.

Fisher (1991) emphasized the importance of discriminating between computer experience (which he placed on a novice-experienced dimension) and domain expertise (which he placed on a naïve-expert dimension). LaLomia and Sidowski (1990) reviewed the scales and questionnaires developed to assess computer satisfaction, literacy and aptitudes. None of the instruments they surveyed specifically addressed measurement of computer experience. Miller et al. (1997) published the Windows Computer Experience Questionnaire (WCEQ), an instrument specifically designed to measure a person's experience with Windows 3.1. The questionnaire took about five minutes to complete and was reliable (coefficient alpha = .74; test-retest correlation = .97). They found that their questionnaire was sensitive to three experiential factors: general Windows experience, advanced Windows experience, and instruction. Smith et al. (1999) distinguished between subjective and objective computer experience. The paper was relatively theoretical and "challenges researchers to devise a reliable and valid measure" (p. 239) for subjective computer experience, but did not offer one.

One user characteristic not addressed in any of the cited literature is one that becomes very important when designing products for international use – cultural characteristics. For example, it is extremely important that in adapting an interface for use by members of another country that all text is accurately translated. It is also important to be sensitive to the possibility that these types of individual differences might be more likely than others to result in crossed interactions.

For comparison studies, having multiple groups (for example, males and females or experts and novices) allows the assessment of potential interactions that might otherwise go unnoticed. Ultimately, the decision for one or multiple groups must be based on expert judgment and a few guidelines. For example, practitioners should consider sampling from different groups if they have reason to believe:

- There are potential and important differences among groups on key measures (Dickens, 1987)
- There are potential interactions as a function of group (Aykin and Aykin, 1991)

- The variability of key measures differs as a function of group
- The cost of sampling differs significantly from group to group

Gordon and Langmaid (1988) recommended the following approach to defining groups:

1. Write down all the important variables.
2. If necessary, prioritize the list.
3. Design an ideal sample.
4. Apply common sense to collapse cells.

For example, suppose a practitioner starts with 24 cells, based on the factorial combination of six demographic locations, two levels of experience, and the two levels of gender. The practitioner should ask himself or herself whether there is a high likelihood of learning anything new and important after completing the first few cells, or would additional testing be wasteful? Can one learn just as much from having one or a few cells that are homogeneous within cells and heterogeneous between cells with respect to an important variable, but are heterogeneous within cells with regard to other, less important variables? For example, a practitioner might plan to (1) include equal numbers of males and females over and under 40 years of age in each cell, (2) have separate cells for novice and experienced users, and (3) drop intermediate users from the test. The resulting design requires testing only two cells (groups), but a design that did not combine genders and age groups in the cells would have required eight cells.

The final issue is the number of participants to include in the test. According to Dumas and Redish (1999), typical usability tests have 6 to 12 participants divided among two to three subgroups. For any given test, the required sample size depends on the number of subgroups, available resources (time/money), and the purpose of the test (for example, precise measurement versus problem discovery). It also depends on whether a study is single-shot (needing a larger sample size) or iterative (needing a smaller sample size per iteration, building up the total sample size over iterations). For a more detailed treatment of this topic, see the section below on sample size estimation.

Test Task Scenarios

As with participants, the most important consideration for test tasks is that they are representative of the types of tasks real users will perform with the product. For any product, there will be a core set of tasks that anyone using the product will perform. People who use barbecue grills use them to cook. People who use desktop speech dictation products use them to produce text. For usability tests, these are the most important tasks to test.

After defining these core tasks, the next step is to list any more peripheral tasks that the test should cover. If a barbecue grill has an external burner for heating pans, it might make sense to include a task that requires participants to work with that burner. If, in addition to the basic vocabulary in a speech dictation system, the program allows users to enable additional special topic vocabularies such as cooking or sports, then it might make sense to devise a task that requires participants to activate and use one of these topics. Practitioners should avoid frivolous or humorous tasks because what is humorous to one person might be offensive to another.

From the list of test tasks, create scenarios of use (with specific goals) that require participants to perform the identified tasks. Critical tasks can appear in more than one scenario. For repeated tasks, vary the task details to increase the generalizability of the results. When testing relatively complex systems, some scenarios should stay within specific parts of the system (for example, typing and formatting a document) and others should explore usage across different parts of the system (for example, creating a figure using a spreadsheet program, adding it to the document, attaching the document to a note, and sending it to a specified recipient).

The complete specification of a scenario should include several items. It is important to document (but not to share with the participant), the required initial conditions so it will be easy to determine before a

test session starts that the system is ready. The written description of the scenario (presented to the participant) should state what the participant is trying to achieve and why (the motivation), keeping the description of the scenario as short as possible to keep the test session moving quickly. The scenario should end with an instruction for the action the participant should take upon finishing the task (to make it easier to measure task completion times). The descriptions of the scenario's tasks should not typically provide step-by-step instructions on how to complete the task, but should include details (for example, actual names and data) rather than general statements. The order in which participants complete scenarios should reflect the way in which users would typically work and with the importance of the scenario, with important scenarios done first unless there are other less important scenarios that produce outputs that the important scenario requires as an initial condition. Not all participants need to receive the same scenarios, especially if there are different groups under study. The tasks performed by administrators of a web system that manages subscriptions will be different from the tasks performed by users who are requesting subscriptions.

Here are some examples of scenarios:

Example 1: "Frank Smith's business telephone number has changed to (896) 555-1234. Please change the appropriate address book entry so you have this new phone number available when you need it. When you have finished, please say 'I'm done.'"

Example 2: "You've just found out that you need to cancel a car reservation that you made for next Wednesday. Please call the system that you used to make the reservation (1-888-555-1234) and cancel it. When you have finished, please hang up the phone and say 'I'm done.'"

Procedure

The test plan should include a description of the procedures to follow when conducting a test session. Most test sessions include an introduction, task performance, post-task activities, and debriefing.

A common structure for the introduction is for the briefer (see the section above on testing roles) to start with the purpose of the test, emphasizing that its goal is to improve the product, not to test the participant. Participation is voluntary, and the participant can stop at any time without penalty. The briefer should inform the participant that all test results will be confidential. The participant should be aware of any planned audio or video recording. Finally, the briefer should provide any special instructions (for example, think-aloud instructions) and answer any other questions that the participant might have.

The participant should then complete any preliminary questionnaires and forms, such as a background questionnaire, an informed consent form (including consent for any recording, if applicable), and, if necessary, a confidential disclosure form. If the participant will be using a workstation, the briefer should help the participant make any necessary adjustments (unless, of course, the purpose of the test is to evaluate workstation adjustability). Finally, the participant should complete any prerequisite training. This can be especially important if the goal of the study is to investigate usability after some period of use (ease of use) rather than immediate usability (ease of learning).

The procedure section should indicate the order in which participants will complete task scenarios. For each participant, start with the first assigned task scenario and complete additional scenarios until the participant finishes (or runs out of time). The procedure section should specify when and how to interact with participants, according to the type of study. This section should also indicate when it is permissible to provide assistance to participants if they encounter difficulties in task performance.

Normally, practitioners should avoid offering assistance unless the participant is visibly distressed. When participants initially request help at a given step in a task, refer them to documentation or other supporting materials if available. If that doesn't help, then provide the minimal assistance required to keep the participant moving forward in the task, note the assistance, and score the task as failed. When participants ask questions, try to avoid direct answers, instead turning their attention back to the task and

encouraging them to take whatever action seems right at that time. When asking questions of participants, it is important to avoid biasing the participant's response. Try to avoid the use of loaded adjectives and adverbs in questions (Dumas and Redish, 1999). Instead of asking if a task was easy, ask the participant to describe what it was like performing the task. Give a short satisfaction questionnaire (such as the ASQ – see the section on questionnaires for details) at the end of each scenario.

After participants have finished the assigned scenarios, it is common to have them complete a final questionnaire, usually a standard questionnaire and any additional items required to cover other test- or product-specific issues. For standardized questionnaires, ISO lists the SUMI (Software Usability Measurement Inventory – Kirakowski, 1996; Kirakowski and Corbett, 1993) and PSSUQ (Post-Study System Usability Questionnaire – Lewis, 1995, 2002). In addition to the SUMI and PSSUQ, ANSI lists the QUIS (Questionnaire for User Interaction Satisfaction – Chin et al., 1988) and SUS (System Usability Scale – Brooke, 1996) as widely used questionnaires. After completing the final questionnaire, the briefer should debrief the participant. Toward the end of debriefing, the briefer should tell the participant that the test session has turned up several opportunities for product improvement (this is almost always true), and thank the participant for his or her contribution to product improvement. Finally, the briefer should discuss any questions the participant has about the test session, and then take care of any remaining activities, such as completing time cards. If there has been any deception employed in the test (which is rare, but can legitimately happen when conducting certain types of simulations), the briefer must inform the participant.

Pilot Testing

Practitioners should always plan for a pilot test before running a usability test. A usability test is a designed artifact, and like any other designed artifact needs at least some usability testing to find problems in the test procedures and materials. A common strategy is to have an initial walkthrough with a member of the usability test team or some other convenient participant. After making the appropriate adjustments, the next pilot participant should be a more representative participant. If there are no changes made to the design of the usability test after running this participant, then the second pilot participant can become the first real participant (but this is rare). Pilot testing should continue until the test procedures and materials have become stable.

Number of Iterations

It is better to run one usability test than not to run any at all. On the other hand, "usability testing is most powerful and most effective when implemented as part of an iterative product development process" (Rubin, 1994, p. 30). Ideally, usability testing should begin early and occur repeatedly throughout the development cycle. When development cycles are short, it is a common practice to run, at a minimum, exploratory usability tests on prototypes at the beginning of a project, to run a usability test on an early version of the product during the later part of functional testing, and then to run another during system testing. Once the final version of the product is available, some organizations run an additional usability test focused on the measurement of usability performance benchmarks. At this stage of development, it is too late to apply information about any problems discovered during the usability test to the soon-to-be-released version of the product, but the information can be useful as early input to a follow-on product if the organization plans to develop another version of the product.

Ethical Treatment of Test Participants

Usability testing always involves human participants, so usability practitioners must be aware of professional practices in the ethical treatment of test participants. Practitioners with professional education in experimental psychology are usually familiar with the guidelines of the American Psychology Association (APA, see http://www.apa.org/ethics/), and those with training in human factors engineering are usually familiar with the guidelines of the Human Factors and Ergonomics Society (HFES, see http://www.hfes.org/About/Code.html). It is particularly important (Dumas, 2003) to be aware of the concepts of informed consent (participants are aware of what will happen during the test, agree to participate, and can leave the test at any time without penalty) and minimal risk (participating in the test does not place participants at any greater risk of harm or discomfort than situations normally encountered in daily life). Most usability tests are consistent with guidelines for informed consent and minimal risk. Only

the test administrator should be able to match a participant's name and data, and the names of test participants should be confidential. Anyone interacting with a participant in a usability test has a responsibility to treat the participant with respect.

Usability practitioners rarely use deception in usability tests. One technique in which there is potential use of deception is the Wizard of Oz (WOZ) method (originally, the OZ Paradigm, Kelley, 1985, also see http://www.musicman.net/oz.html). In a test using the WOZ method, a human (the Wizard) plays the part of the system, remotely controlling what the participant sees happen in response to the participant's manipulations. This method is particularly effective in early tests of speech recognition interactive voice response (IVR) systems because all the Wizard needs is a script and a phone (Sadowski, 2001). Often, there is no compelling reason to deceive participants, so they know that the system they are working with is remotely controlled by another person for the purpose of early evaluation. If there is a compelling need for deception (for example, to manage the participant's expectations and encourage natural behaviors), then this deception must be revealed to the participant during debriefing.

Reporting Results

There are two broad classes of usability test results, problem reports and quantitative measurements. It is possible for a test report to contain one type exclusively (for example, the ANSI Common Industry Format has no provision for reporting problems), but most usability test reports will contain both types of results.

Describing Usability Problems

"We broadly define a usability defect as: Anything in the product that prevents a target user from achieving a target task with reasonable effort and within a reasonable time. … Finding usability problems is relatively easy. However, it is much harder to agree on their importance, their causes and the changes that should be made to eliminate them (the fixes)." (Marshall et al., 1990, p. 245)

The best way to describe usability problems depends on the purpose of the descriptions. For usability practitioners, the goal should be to describe problems in such a way that the description leads logically to one or more potential interventions (recommendations). Ideally, the problem description should also include some indication of the importance of fixing the problem (most often referred to as problem severity). For more scientific investigations, there can be value in higher levels of problem description (Keenan et al., 1999), but developers rarely care about these levels of description. They just want to know what they need to do to make things better while also managing the cost (both monetary and time) of interventions (Gray and Salzman, 1998).

The problem description scheme of Lewis and Norman (1986) has both scientific and practical merit because their problem description categories indicate, at least roughly, an appropriate intervention. They stated (p. 413) that "although we do not believe it possible to design systems in which people do not make errors, we do believe that much can be done to minimize the incidence of error, to maximize the discovery of the error, and to make it easier to recover from the error." They separated errors into mistakes (errors due to incorrect intention) and slips (errors due to appropriate intention but incorrect action), further breaking slips down into mode errors (which indicate a need for better feedback or elimination of the mode), capture errors (which indicate a need for better feedback), and description errors (which indicate a need for better design consistency). In one study using this type of problem categorization, Prümper et al. (1992) found that expertise did not affect the raw number of errors made by participants in their study, but experts handled errors much more quickly than novices. The types of errors that experts made were different from those made by novices, with experts' errors primarily occurring at the level of slips rather than mistakes (knowledge errors).

Rasmussen (1986), using an approach similar to that of Lewis and Norman (1986), described three levels of errors: skill-based, rule-based, and knowledge-based. Two relatively new classification schemes are Structured Usability Problem EXtraction, or SUPEX (Cockton and Lavery, 1999) and the User Action

Framework, or UAF (Andre et al., 2000). The UAF requires a series of decisions, starting with an Interaction Cycle (Planning, Physical Actions, Assessment) based on the work of Norman (1986). Most classifications require four or five decisions, with inter-rater reliability (as measured with kappa) highest at the first step (κ=.978), but remaining high through the fourth and fifth steps (κ>.7).

Whether any of these classification schemes will see widespread use by usability practitioners is still unknown. There is considerable pressure on practitioners to produce results and recommendations as quickly as possible. Even if these classification schemes see little use by practitioners, effective problem classification is a very important problem to solve as usability researchers strive to compare and improve usability testing methods.

Crafting Design Recommendations from Problem Descriptions

As indicated by the title of this section, the development of recommendations from problem descriptions is a craft rather than a rote procedure. A well-written problem description will often strongly imply an intervention, but it is also often the case that there might be several ways to attack a problem. It can be helpful for practitioners to discuss problems and potential interventions with the other members of their team, and to get input from other stakeholders as necessary (especially, the developers of the product). This is especially important if the practitioner has observed problems but is uncertain as to the appropriate level of description of the problem.

For example, suppose you have written a problem description about a missing Help button in a software application. This could be a problem with the overall design of the software, or might be a problem isolated to one screen. You might be able to determine this by inspecting other screens in the software, but it could be faster to check with one of the developers.

The first recommendations to consider should be for interventions that will have the widest impact on the product. "Global changes affect everything and need to be considered first." (Rubin, 1994, p. 285) After addressing global problems, continue working through the problem list until there is at least one recommendation for each problem. For each problem, start with interventions that would eliminate the problem, then follow, if necessary, with other less drastic (less expensive, more likely to be implemented) interventions that would reduce the severity of the remaining usability problem. When different interventions involve different tradeoffs, it is important to communicate this clearly in the recommendations. This approach can lead to two tiers of recommendations – those that will happen for the version of the product currently under development (short-term) and those that will happen for a future version of the product (long-term).

Prioritizing Problems

Because usability tests can reveal more problems than there are resources to address, it is important to have some means for prioritizing problems. There are two approaches to prioritization that have appeared in the usability testing literature: (1) judgment driven (Virzi, 1992) and (2) data driven (Dumas and Redish, 1999; Lewis et al., 1990; Rubin, 1994). The bases for judgment-driven prioritizations are the ratings of stakeholders in the project (such as usability practitioners and developers). The bases for data-driven prioritizations are the data associated with the problems, such as frequency, impact, ease of correction, and likelihood of usage of the portion of the product that was in use when the problem occurred. Of these, the most common measurements are frequency and impact (sometimes referred to as severity, although, strictly speaking, severity should include the effect of all of the types of data considered for prioritization). Hassenzahl (2000), in a study of the two approaches to prioritization, found a lack of correspondence between data-driven and judgment-driven severity estimates. This suggests that the preferred approach should be data-driven.

The usual method for measuring the frequency of occurrence of a problem is to divide the number of occurrences within participants by the number of participants. A common method (Dumas and Redish, 1999; Rubin, 1994) for assessing the impact of a problem is to assign impact scores according to whether the problem (1) prevents task completion, (2) causes a significant delay or frustration, (3) has a relatively minor

effect on task performance, or (4) is a suggestion. This is similar to the scheme of Lewis et al. (1990), in which the impact levels were (1) scenario failure or irretrievable data loss (for example, the participant required assistance to get past the problem or caused the participant to believe the scenario to be properly completed when it wasn't), (2) considerable recovery effort (recovery took more than one minute or the participant repeatedly experienced the problem within a scenario), (3) minor recovery effort (the problem occurred only once within a scenario with recovery time at or under one minute), or (4) inefficiency (a problem not meeting any of the other criteria).

When considering multiple types of data in a prioritization process, it is necessary to combine the data in some way. A graphical approach is to create a problem grid with frequency on one axis and impact on the other (see Figure 3). High-frequency high-impact problems would receive treatment before low-frequency low-impact problems. The relative treatment of high-frequency low-impact problems and low-frequency high-impact problems depends on practitioner judgment.

An alternative approach is to combine the data arithmetically. Rubin (1994) described a procedure for combining four levels of impact (using the criteria described above with 4 assigned to the most serious level) with four levels of frequency (4: frequency = 90%; 3: 51-89%; 2: 11-50%; 1: = 10%) by adding the scores. For example, if a problem had an observed frequency of occurrence of 80% and had a minor effect on performance, then its priority would be 5 (a frequency rating of 3 plus an impact rating of 2). With this approach, priority scores can range from a low of 2 to a high of 8. If information is available about the likelihood that a user would work with the part of the product that enables the problem, then this information would be used to adjust the frequency rating. Continuing the example, if the expectation is that only 10% of users would encounter the problem, then the priority would be 3 (a frequency rating of 1 for the 10% x 80%, or 8% likelihood of occurrence plus an impact rating of 2).

A similar strategy is to multiply the observed percentage frequency of occurrence by the impact score. The range of priorities depends on the values assigned to each impact level. Assigning 10 to the most serious impact level leads to a maximum priority (severity) score of 1000 (which can optionally be divided by 10 to create a scale that ranges from 1 to 100). Appropriate values for the remaining three impact categories depend on practitioner judgment, but a reasonable set is 5, 3, and 1. Using those values, the problem with an observed frequency of occurrence of 80% and a minor effect on performance would have a priority of 24 (80 x 3/10). It is possible to extend this method to account for likelihood of use using the same procedure as that described by Rubin (1994), which in the example resulted in modifying the frequency measurement from 80% to 8%. Another way to extend the method is to categorize the likelihood of use with a set of categories such as very high likelihood (assigned a score of 10), high likelihood (assigned a score of 5), moderate likelihood (assigned a score of 3), and low likelihood (assigned a score of 1), and multiplying all three scores to get the final priority (severity) score (then optionally divide by 100 to create a scale that ranges from 1 to 100). Continuing the previous example with the assumption that the task in which the problem occurred has a high likelihood of occurrence, the problem's priority would be 12 (5 x 240/100). In most cases, applying the different data-driven prioritization schemes to the same set of problems should result in a very similar prioritization.

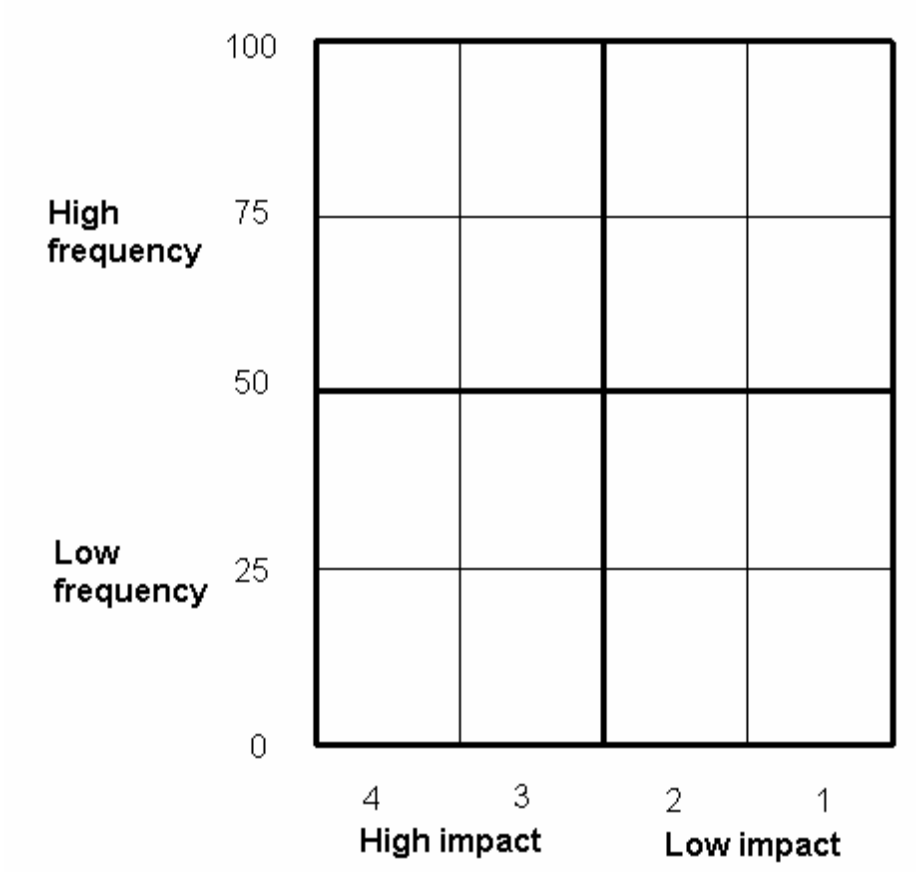

Figure 3. Sample problem grid.

Working with Quantitative Measurements

The most common use of quantitative measurements is to characterize performance and preference variables by computing means, standard deviations, and, ideally, confidence intervals. Practitioners use these results to compare observed to target measurements when targets are available. When targets are not available, the results can still be informative, for example, for use as future target measurements or as relatively gross diagnostic indicators.

The failure to meet targets is an obvious diagnostic cue. A less obvious cue is an unusually large standard deviation. Landauer (1997) describes a case in which the times to record an order were highly variable. The cause for the excessive variability was that a required phone number was sometimes, but not always, available, which turned out to be an easy problem to fix. Because the means and standard deviations of time scores tend to correlate, one way to detect an unusually large variance is to compute the coefficient of variation by dividing the standard deviation by the mean (Jeff Sauro, personal communication, April 26, 2004) or the normalized performance ratio by dividing the mean by the standard deviation (Moffat, 1990). Large coefficients of variation (or, correspondingly, small normalized performance ratios) are potentially indicative of the presence of usability problems.

ADVANCED TOPICS

This section covers more advanced topics in usability testing, including sample size estimation for problem discovery and measurement tests (both comparative and parameter estimation), confidence intervals based on *t*-scores and binomial confidence intervals, and standardized usability questionnaires. This chapter contains a considerable amount of information about statistical topics because statistical methods do not typically receive much attention in chapters on usability testing and, properly practiced, these techniques can be very valuable. On the other hand, practitioners should keep in mind that the most important factors that lead to successful usability evaluation are the appropriate selection of participants and tasks. No statistical analysis can repair a study in which you watch the wrong people doing the wrong activities.

Sample Size Estimation

The purpose of this section is to discuss the principles of sample size estimation for three types of usability test: population parameter estimation, comparative (also referred to as experimental), and problem-discovery (also referred to as diagnostic, observational, or formative). This section assumes some knowledge of introductory applied statistics, so if you're not comfortable with terms such as mean, variance, standard deviation, *p*, *t*-score, and *Z*-score, refer to an introductory statistics text such as Walpole (1976) for definitions of these and other fundamental terms.

Sample size estimation requires a blend of mathematics and judgment. The computations are straightforward, and it is possible to make reasoned judgments (for example, judgments about expected costs and precision requirements) for those values that the mathematics cannot determine.

Sample Size Estimation for Parameter Estimation and Comparative Studies

Traditional sample size estimation for population parameter estimation and comparative studies depends on having an estimate of the variance of the dependent measure(s) of interest and an idea of how precise (the magnitude of the critical difference and the statistical confidence level) the measurement must be (Walpole, 1976). Once you have that, the rest is mathematical mechanics (typically using the formula for the *t* statistic.)

You can (1) get an estimate of variance from previous studies using the same method (same or similar tasks and measures), (2) you can run a quick pilot study to get the estimate (for example, piloting with four participants should suffice to provide an initial estimate of variability), or (3) you can set the critical difference your are trying to detect to some fraction of the standard deviation (Diamond, 1981). (See the following examples for more details about these different methods).

Certainly, people prefer precise measurement to imprecise measurement, but all other things being equal, the more precise a measurement is, the more it will cost, and running more participants than necessary is wasteful of resources (Kraemer and Thiemann, 1987). The process of carrying out sample size estimation can also lead usability practitioners and their management to a realistic determination of how much precision they really need to make their required decisions.

Alreck and Settle (1985) recommend using a 'what if' approach to help decision makers determine their required precision. Start by asking the decision maker what would happen if the average value from the study was off the true value by one percent. Usually, the response would be that a difference that small wouldn't matter. Then ask what would happen if the measurement were off by five percent. Continue until you determine the magnitude of the critical difference. Then start the process again, this time pinning down the required level of statistical confidence. Note that statistically unsophisticated decision makers are likely to start out by expecting 100% confidence (which is only possible by sampling every unit in the population). Presenting them with the sample sizes required to achieve different levels of confidence can help them settle in on a more realistic confidence level.

Example 1: Parameter estimation given estimate of variability and realistic criteria. The following example illustrates the process of computing the sample size requirement for the estimation of a population parameter given an existing estimate of variability and realistic measurement criteria. For speech recognition applications, the recognition accuracy is an important value to track due to the adverse effects misrecognitions have on product usability. Thus, part of the process of evaluating the usability of a speech recognition product is estimating its accuracy. For this example, suppose:

- Recognition variability (variance) from a previous similar evaluation = 6.35
- Critical difference (d) = 2.5%
- Desired level of confidence: 90%

The appropriate procedure for estimating a population parameter is to construct a confidence interval (Bradley, 1976). To determine the upper and lower limits of a confidence interval, add to and subtract the following from the observed mean:

[1] sem * t_{crit}

where sem is the standard error of the mean (the standard deviation, S, divided by the square root of the sample size, n) and t_{crit} is the t-value associated with the desired level of confidence (found in a t-table, available in most statistics texts). Setting the critical difference to 2.5 is the same as saying that the value of sem * t_{crit} should be equal to 2.5. In other words, you don't want the upper or lower bound of the confidence interval to be more than 2.5 percentage points away from the observed mean, for a confidence interval width equal to 5.0.

Calculating the sem depends on knowing the sample size, and the value of t_{crit} also depends on the sample size, but you don't know the sample size yet. Iterate using the following method.

1. Start with the Z-score for the desired level of confidence in place of t_{crit}. For 90% confidence, this is 1.645. (By the way, if you actually **know** the true variability for the measurement rather than just having an estimate, you're done at this point because it's appropriate to use the Z-score rather than a t-score. However, you almost never know the true variability, but must work with estimates.)
2. Algebraic manipulations based on the formula sem * $Z = d$ results in $n = (Z^2 * S^2)/d^2$ which, for this example, is $n = (1.645^2 * 6.35)/2.5^2$, which equals 2.7. Always round sample size estimates up to the next whole number, so this initial estimate is 3.
3. Now you need to adjust the estimate by replacing the Z-score with the t-score for a sample size of 3. For this estimate, the degrees of freedom (df) to use when looking up the value in a t table is n-1, or 2. This is important because the value of Z will always be smaller than the appropriate value of t, making the initial estimate smaller than it should be. For this example, t_{crit} is 2.92.
4. Recalculating for n using 2.92 in place of 1.645 produces 8.66, which rounds up to 9.
5. Because the appropriate value of t_{crit} is now a little smaller than 2.92 (because the estimated sample size is now larger, with 9-1 or 8 degrees of freedom), recalculate n again, using t_{crit} equal to 1.860. The new value for n is 3.515, which rounds up to 4.
6. Stop iterating when you get the same value for n on two iterations or you begin cycling between two values for n, in which case you should choose the larger value. Table 2 shows the full set of iterations for this example, which ends by estimating the appropriate sample size as 5.

Table 2. Full set of iterations for Example 1

	Iteration					
	Initial	**1**	**2**	**3**	**4**	**5**
t_{crit}	1.645	2.92	1.86	2.353	2.015	2.132
t_{crit}^2	2.71	8.53	3.46	5.54	4.06	4.55
S^2	6.35	6.35	6.35	6.35	6.35	6.35
d	2.5	2.5	2.5	2.5	2.5	2.5
Estimated n	2.7493	8.663	3.515	5.6252	4.1252	4.618
Rounded up	3	9	4	6	5	5
df	2	8	3	5	4	4

Note that there is nothing in these computations that makes reference to the size of the population. Unless the size of the sample is a significant percentage of the total population under study (which is rare, but correctable using a finite population correction), the size of the population is irrelevant. Alreck and Settle (1985) explain this with a soup-tasting analogy. Suppose you're cooking soup in a one-quart saucepan, and want to test if it's hot enough. You would stir it thoroughly, then taste one teaspoon. If it were a two-quart saucepan, you'd follow the same procedure – stir thoroughly, then taste one teaspoon.

Diamond (1981) points out that you can usually get by with an initial estimate and one iteration because most researchers don't mind having a sample size that's a little larger than necessary. If the cost of each sample is high, though, it makes sense to iterate until reaching one of the stopping criteria. Note that the initial estimate establishes the lower bound for the sample size (3 in this example), and the first iteration establishes the upper bound (9 in this example).

Example 2: Parameter estimation given estimate of variability and unrealistic criteria. The measurement criteria in Example 1 were reasonable – 90% confidence that the interval (limited to a total length of 5%) contains the true mean. The next example (Example 2) shows what happens when the measurement criteria are less realistic, illustrating the potential cost associated with high confidence and high measurement precision. Suppose the measurement criteria for the situation described in Example 1 were less realistic, with:

- Recognition variability from a previous similar evaluation = 6.35
- Critical difference (*d*) = .5%
- Desired level of confidence: 99%

In that case, the initial *Z*-score would be 2.576, and the initial estimate of *n* would be:

[2] $n = (2.576^2 * 6.35)/.5^2 = 168.549$ (which rounds up to 169).

Recalculating *n* with t_{crit} equal to 2.605 (*t* with 168 degrees of freedom) results in *n* equal to 172.37, which rounds up to 173. (Rather then continuing to iterate, note that the final value for the sample size must lie between 169 and 173.) There might be some industrial environments in which usability investigators would consider 169 to 173 participants a reasonable and practical sample size, but they are rare. (On the other hand, collecting data from this number of participants or more in a mailed survey is common.)

Example 3: Parameter estimation given no estimate of variability. For both Examples 1 and 2, it doesn't matter if the estimate of variability came from a previous study or a quick pilot study. Suppose, however, that you don't have any idea what the measurement variability is, and it's too expensive to run a

pilot study to get an initial estimate. Example 3 illustrates a technique (from Diamond, 1981) for getting around this problem. To do this, though, you need to give up a definition of the critical difference (*d*) in terms of the variable of interest and replace it with a definition in terms of a fraction of the standard deviation.

In this example, the measurement variance is unknown. To get started, the testers have decided that, with 90% confidence, they do not want *d* to exceed half the value of the standard deviation. The measurement criteria are:

- Recognition variability from a previous similar evaluation = N/A
- Critical difference (*d*) = .5*S*
- Desired level of confidence: 90%

The initial sample size estimate is:

[3] $n = (1.645^2 * S^2)/(.5S)^2 = (1.645^2)/(.5)^2 = 10.824$, which rounds up to 11.

The result of the first iteration, replacing 1.645 with t_{crit} for 10 degrees of freedom (1.812), results in a sample size estimation of 13.13, which rounds up to 14. The appropriate sample size is therefore somewhere between 11 and 14, with the final estimate determined by completing the full set of iterations.

Example 4: Comparing a parameter to a criterion. For an example comparing a measured parameter to a criterion value, suppose that you have a product requirement that installation should take no more than 30 minutes. In a preliminary evaluation, participants needed an average of 45 minutes to complete installation. Development has fixed a number of usability problems found in that preliminary study, so you're ready to measure installation time again, using the following measurement criteria:

- Performance variability from the previous evaluation = 10.0
- Critical difference (*d*) = 3 minutes
- Desired level of confidence: 90%

The interpretation of these measurement criteria is that we want to be 90% confident that we can detect a difference as small as 3 minutes between the mean of the data we gather in the test and the criterion we're trying to beat. In other words, the installation will pass if the observed mean time is 27 minutes or less, because the sample size should guarantee an upper limit to the confidence interval that is no more than 3 minutes above the mean (as long as the observed variance is less than or equal to the initial estimate of the variance). The procedure for determining the sample size in this situation is the same as that of Example 1, shown in Table 3. The outcome of these iterations is a sample size requirement of 6 because the sample size estimates begin cycling between 5 and 6.

Table 3. Full set of iterations for Example 4

	Initial	1	2	3	4
t_{crit}	1.645	2.353	1.943	2.132	2.015
t_{crit}^2	2.706	5.537	3.775	4.545	4.060
s^2	10	10	10	10	10
d	3	3	3	3	3
d^2	9	9	9	9	9
Estimated n	3.006694	6.151788	4.194721	5.050471	4.511361
Rounded up	4	7	5	6	5
df	3	6	4	5	4

Example 5: Sample size for a paired t-test. When you obtain two comparable measurements from each participant in a test (a within-subjects design), you can assess the results using a paired *t*-test. Another name for a ""paired *t*-test" is a "difference score *t*-test", because the measurements of concern are the mean and standard deviation of the set of difference scores rather than the raw scores. Suppose you plan to obtain recognition accuracy scores from participants who have dictated test texts into your product under development and a competitor's product (following all the appropriate experimental design procedures such as counterbalancing the order of presentation of products to participants – see a text such as Myers, 1979 for guidance in experimental design), using the following criteria:

- Difference score variability from a previous evaluation = 5.0
- Critical difference (*d*) = 2%
- Desired level of confidence: 90%

This situation is similar to that of the previous example, because the typical goal of a difference scores *t*-test is to determine if the average difference between the scores is statistically significantly different from 0. Thus, the usability criterion in this case is 0, and we want to be 90% confident that if the true difference between the systems' accuracies is 2% or more, then we will be able to detect it because the confidence interval for the difference scores will not contain 0. Table 4 shows the iterations for this situation, leading to a sample size estimate of 6.

Table 4. Full set of iterations for Example 5

	Initial	1	2	3	4
t_{crit}	1.645	2.353	1.943	2.132	2.015
t_{crit}^2	2.706	5.537	3.775	4.545	4.060
s^2	5	5	5	5	5
d	2	2	2	2	2
d^2	4	4	4	4	4
Estimated n	3.382531	6.920761	4.719061	5.68178	5.075281
Rounded up	4	7	5	6	6
df	3	6	4	5	5

Example 6: Sample size for a two-groups t-test. Up to this point, the examples have all involved one group of scores, and have been amenable to similar treatment. If you have a situation in which you plan

to compare scores from two independent groups, then things get a little more complicated. For one thing, you now have two sample sizes to consider – one for each group.

To simplify things in this example, assume that the groups are essentially equal (especially with regard to performance variability), which should be the case if the groups contain participants from a single population who have received random assignment to treatment conditions. In this case, it is reasonable to believe that the sample size for both groups will be equal, which simplifies things. For this situation, the formula for the initial estimate of the sample size for each group is:

[4] $n = (2*Z^2*S^2)/d^2$

Note that this is the similar to the formula presented in Example 1, with the numerator multiplied by 2. After getting the initial estimate, begin iterating using the appropriate value for t_{crit} in place of *Z*. For example, suppose we needed to conduct the experiment described in Example 5 with independent groups of participants, keeping the measurement criteria the same:

- Estimate of variability from a previous evaluation = 5.0
- Critical difference (*d*) = 2%
- Desired level of confidence: 90%

In that case, iterations would converge on a sample size of 9 participants per group, for a total sample size of 18, as shown in Table 5.

Table 5. Full set of iterations for Example 6

	Initial	**1**	**2**	**3**
t_{crit}	1.645	1.943	1.833	1.86
t_{crit}^2	2.706	3.775	3.360	3.460
s^2	5	5	5	5
d	2	2	2	2
d^2	4	4	4	4
Estimated n	6.765	9.438	8.400	8.649
Rounded up	7	10	9	9
df	6	9	8	8

This illustrates the well-known measurement efficiency of experiments that produce difference scores (within-subjects designs) relative to experiments involving independent groups (between-subjects designs). For the same measurement precision, the estimated sample size for Example 5 was six participants, 1/3 the sample size requirement estimated for Example 6.

Doing this type of analysis gets more complicated if you have reason to believe that the groups are different, especially with regard to variability of performance. In that case, you would want to have a larger sample size for the group with greater performance variability in an attempt to obtain more equal precision of measurement for each group. Advanced market research texts (such as Brown, 1980) provide sample size formulas for these situations.

Example 7: Making power explicit in the sample size formula. The power of a procedure is not an issue when estimating the value of a parameter, but it is an issue when testing a hypothesis (as in Example 6). In traditional hypothesis testing, there is a null (H_o) and an alternative (H_a) hypothesis. The typical null hypothesis is that there is no difference between groups, and the typical alternative hypothesis is that the difference is greater than zero. When the alternative hypothesis is that the difference is nonzero, the test is

two-tailed because you can reject the null hypothesis with either a sufficiently positive or a sufficiently negative outcome. If you have reason to believe that you can predict the direction of the outcome, or if an outcome in only one direction is meaningful, you can construct an alternative hypothesis that considers only a sufficiently positive or a sufficiently negative outcome (a one-tailed test). For more information, see an introductory statistics text (such as Walpole, 1976).

When you test a hypothesis (for example, that the difference in recognition accuracy between two competitive dictation products is nonzero), there are two ways to make a correct decision and two ways to be wrong, as shown in Table 6.

Table 6. Possible outcomes of a hypothesis test

	Reality	
Decision	*H_o is true*	*H_o is false*
Insufficient evidence to reject H_o	Fail to reject H_o	Type II error
Sufficient evidence to reject H_o	Type I error	Reject H_o

Strictly speaking, you never accept the null hypothesis, because the failure to acquire sufficient evidence to reject the null hypothesis could be due to (1) no significant difference between groups or (2) a sample size too small to detect an existing difference. Rather than accepting the null hypothesis, you fail to reject it.

Returning to Table 6, the two ways to be right are (1) to fail to reject the null hypothesis (H_0) when it is true or (2) to reject the null hypothesis when it is false. The two ways to be wrong are (1) to reject the null hypothesis when it is true (Type I error) or (2) to fail to reject the null hypothesis when it is false (Type II error).

Table 7 shows the relationship between these concepts and their corresponding statistical testing terms:

Table 7. Statistical testing terms

Statistical Concept	**Testing Term**
Acceptable probability of a Type I error	Alpha
Acceptable probability of a Type II error	Beta
Confidence	1-alpha
Power	1-beta

The formula presented in Example 6 for an initial sample size estimate was:

[5] $n = (2*Z^2*S^2)/d^2$

In Example 6, the *Z*-score was set for 90% confidence (which means alpha = .10). To take power into account in this formula, you need to add another *Z*-score to the formula – the *Z*-score associated with the desired power of the test (as defined in Table 7). Thus, the formula becomes:

[6] $n = (2*(Z_a + Z_b)^2*S^2)/d^2$

So, what was the value for power in Example 6? When beta equals .5 (in other words, when the power is 50%), the value of z_b is 0, so z_b disappears from the formula. Thus, in Example 6, the implicit power was 50%. Suppose you want to increase the power of the test to 80%, reducing beta to .2.

- Estimate of variability from a previous evaluation = 5.0

- Critical difference (d) = 2%
- Desired level of confidence: 90% (Z_a=1.645)
- Desired power: 80% (Z_b=1.282)

With this change, the iterations converge on a sample size of 24 participants per group, for a total sample size of 48, as shown in Table 8. To achieve the stated goal for power results in a considerably larger sample size.

Table 8. Full set of iterations for Example 7

	Initial	**1**	**2**	**3**
t(alpha)	1.645	1.721	1.714	1.717
t(beta)	1.282	1.323	1.319	1.321
t(total)	2.927	3.044	3.033	3.038
t(total)^2	8.567	9.266	9.199	9.229
S^2	5	5	5	5
d	2	2	2	2
d^2	4	4	4	4
Estimated n	21.418	23.165	22.998	23.074
Rounded up	22	24	23	24
df	21	23	22	23

Note that the stated power of a test is relative to the critical difference – the smallest effect worth finding. Either increasing the value of the critical difference or reducing the power of a test will result in a smaller required sample size.

Appropriate statistical criteria for industrial testing. In scientific publishing, the usual criterion for statistical significance is to set the permissible Type I error (alpha) equal to 0.05. This is equivalent to seeking to have 95% confidence that the effect is real rather than random, and is focused on controlling the Type I error (the likelihood that you decide that an effect is real when it's random). There is no corresponding scientific recommendation for the Type II error (beta, the likelihood that you will conclude an effect is random when it's real), although some suggest setting it to .20 (Diamond, 1981). The rationale behind the emphasis on controlling the Type I error is that it is better to delay the introduction of good information into the scientific database (a Type II error) than to let erroneous information in (a Type I error).

In industrial evaluation, the appropriate values for Type I and Type II errors depend on the demands of the situation – whether the cost of a Type I or Type II error would be more damaging to the organization. Because we are often resource-constrained, especially with regard to making timely decisions to compete in dynamic marketplaces, this paper has used measurement criteria (such as 90% confidence rather than 95% confidence and fairly large values for d) that seek a greater balance between Type I and Type II errors than is typical in work designed to result in scientific publications. Nielsen (1997) has suggested that 80% confidence is appropriate for practical development purposes. For an excellent discussion of this topic for usability researchers, see Wickens (1998). For other technical issues and perspectives, see Landauer (1997).

Another way to look at the issue is to ask the question, "Am I typically interested in small high-variability effects or large low-variability effects?" The correct answer depends on the situation, but in usability testing, the emphasis is on the detection of large low-variability effects (either large performance effects or frequently-occurring problems). You shouldn't need a large sample to verify the existence of large low-variability effects. Some writers equate sample size with population coverage, but this isn't true. A small sample size drawn from the right population provides better coverage than a large sample size drawn

from the wrong population. The statistics involved in computing confidence intervals from small samples compensate for the potentially smaller variance in the small sample by forcing the confidence interval to be wider than that for a larger sample (specifically, the value of *t* is greater when samples are smaller).

Coming from a different tradition than usability research, many market research texts provide rules of thumb recommending large sample sizes. For example, Aaker and Day (1986) recommend a minimum of 100 per group, with 20-50 for subgroups. For national surveys with many subgroup analyses, the typical total sample size is 2500 (Sudman, 1976). These rules of thumb do not make any formal contact with statistical theory, and may in fact be excessive, depending on the goals of the study. Other market researchers (for example, Banks, 1965) do promote a careful evaluation of the goals of a study.

> It is urged that instead of a policy of setting uniform requirements for type I and II errors, regardless of the economic consequences of the various decisions to be made from experimental data, a much more flexible approach be adopted. After all, if a researcher sets himself a policy of always choosing the apparently most effective of a group of alternative treatments on the basis of data from unbiased surveys or experiments and pursues this policy consistently, he will find that in the long run he will be better off than if he chose any other policy. This fact would hold even if none of the differences involved were statistically significant according to our usual standards or even at probability levels of 20 or 30 percent. (Banks, 1965, p. 252)

Finally, Alreck and Settle (1985) provide an excellent summary of the factors indicating appropriate use of large and small samples.

Use a large sample size when:

1. Decisions based on the data will have very serious or costly consequences
2. The sponsors (decision-makers) demand a high level of confidence
3. The important measures have high variance
4. Analyses will require the dividing of the total sample into small subsamples
5. Increasing the sample size has a negligible effect on the cost and timing of the study
6. Time and resources are available to cover the cost of data collection

Use a small sample size when:

1. The data will determine few major commitments or decisions
2. The sponsors (decision-makers) require only rough estimates
3. The important measures have low variance
4. Analyses will use the entire sample, or just a few relatively large subsamples
5. Costs increase dramatically with sample size
6. Budget constraints or time limitations limit the amount of data you can collect

Some tips on reducing variance. Because measurement variance is such an important factor in sample size estimation for these types of studies, it generally makes sense to attempt to manage variance (although in some situations, such management is out of a practitioner's control). Here are some ways to reduce variance:

- Make sure participants understand what they are supposed to do in the study. Unless potential participant confusion is part of the evaluation (and it sometimes is), it can only add to measurement variance.

- One way to accomplish this is through practice trials that allow participants to get used to the experimental situation without unduly revealing study-relevant information.
- If appropriate, use expert rather than novice participants. Almost by definition, expertise implies reduced performance variability (increased automaticity) (Mayer, 1997). With regard to reducing variance, the farther up the learning curve, the better.
- A corollary of this is that if you need to include both expert and novice users, you should be able to get equal measurement precision for both groups with unequal sample sizes (fewer experts required than novices – which is good, because experts are typically harder than novices to recruit as participants).
- If appropriate, study simple rather than complex tasks.
- Use data transformations for measurements that typically exhibit correlations between means and variances or standard deviations. For example, frequency counts often have proportional means and variances (treated with the square root transformation), and time scores often have proportional means and standard deviations (treated with the logarithmic transformation) (Myers, 1979).
- For comparative studies, use within-subjects designs rather than between-subjects designs whenever possible.
- Keep user groups as homogeneous as possible (but although this reduces variability, it can simultaneously pose a threat to a study's external validity if the test group is more homogenous than the population under study) (Campbell and Stanley, 1963).

Keep in mind that it is reasonable to use these tips only when their use does not adversely affect the validity and generalizability of the study. Having a valid and generalizable study is far more important than reducing variability.

Some tips for estimating unknown variance. Parasuraman (1986) described a method for estimating variability if you have an idea about the largest and smallest values for a population of measurements, but don't have the information you need to actually calculate the variability. Estimate the standard deviation (the square root of the variability) by dividing the difference between the largest and smallest values by 6. This technique assumes that the population distribution is normal, and then takes advantage of the fact that 99% of a normal distribution will lie in the range of plus or minus three standard deviations of the mean.

Nielsen (1997) surveyed 36 published usability studies, and found that the mean standard deviation for measures of expert performance was 33% of the mean value of the usability measure (in other words, if the mean completion time was 100 seconds, then the mean standard deviation was about 33 seconds). For novice-user learning the mean standard deviation was 46%, and for measures of error rates the value was 59%.

Churchill (1991) provided a list of typical variances for data obtained from rating scales. Because the number of items in the scale affects the possible variance (with more items leading to more variance), the table takes the number of items into account. For five-point scales, the typical variance is 1.2-2.0; for seven-point scales it is 2.4-4.0; and for ten-point scales it is 3.0-7.0. Because data obtained using rating scales tends to have a more uniform than normal distribution, he advises using a number nearer the high end of the listed range when estimating sample sizes.

Measurement theorists who agree with S. S. Steven's (1951) principle of invariance might yell 'foul' at this point because they believe it is not permissible to calculate averages or variances from rating scale data. There is considerable controversy on this point (for example, see Lord, 1953, Nunnally, 1976, or Harris, 1985). Data reported by Lewis (1993) indicate that taking averages and conducting *t*-tests on multipoint rating data provides far more interpretable and consistent results than the alternative of taking medians and conducting Mann-Whitney *U*-tests. You do have to be careful not to act as if rating scale data are interval data rather than ordinal data when you make claims about the meaning of the outcomes of your statistical tests. An average rating of 4 might be better than an average rating of 2, but you can't claim that it is twice as good (a ratio claim), nor can you claim that the difference between 4 and 2 is equal to the difference between 4 and 6 (an interval claim).

Sample Size Estimation for Problem-Discovery (Formative) Studies

"Having collected data from a few test subjects – and initially a few are all you need – you are ready for a revision of the text." (Al-Awar et al., 1981, p. 34)

"This research does not mean that all of the *possible* problems with a product appear with 5 or 10 participants, but most of the problems that are going to show up with one sample of tasks and one group of participants will occur early." (Dumas, 2003, p. 1098)

While these types of general guidelines have been helpful, it is possible to use more precise methods to estimate sample size requirements for problem-discovery usability tests. Estimating sample sizes for tests that have the primary purpose of discovering the problems in an interface depends on having an estimate of p, characterized as the average likelihood of problem occurrence or, alternatively, the problem discovery rate. As with comparative studies, this estimate can come from previous studies using the same method and similar system under evaluation, or can come from a pilot study. For standard scenario-based usability studies, the literature contains large-sample examples that show p ranging from .16 to .42 (Lewis, 1994). For heuristic evaluations, the reported value of p from large-sample studies ranges from .22 to .60 (Nielsen and Molich, 1990).

When estimating p from a small sample, it is important to adjust its initially estimated value because a small-sample estimate of p (for example, fewer than 20 participants) has a bias that results in potentially substantial overestimation of its value (Hertzum and Jacobsen, 2003). A series of Monte Carlo experiments (Lewis, 2001a) have demonstrated that a formula combining Good-Turing discounting with a normalization procedure provides a reasonably accurate adjustment of initial estimates of p (p_{est}), even when the sample size for that initial estimate has as few as two participants (preferably four participants, though, because the variability of estimates of p is greater for smaller samples, Faulkner, 2003; Lewis, 2001a). This formula for the adjustment of p is:

$$[7]\ p_{adj} = \tfrac{1}{2}[(p_{est} - 1/n)(1 - 1/n)] + \tfrac{1}{2}[\,p_{est}\,/(1+GT_{adj})]$$

where GT_{adj} is the Good-Turing adjustment to probability space (which is the proportion of the number of problems that occurred once divided by the total number of different problems). The $p_{est}/(1+ GT_{adj})$ component in the equation produces the Good-Turing adjusted estimate of p by dividing the observed, unadjusted estimate of p (p_{est}) by the Good-Turing adjustment to probability space. The $(p_{est} - 1/n)(1 - 1/n)$ component in the equation produces the normalized estimate of p from the observed, unadjusted estimate of p and n (the sample size used to estimate p). The reason for averaging these two different estimates is that the Good-Turing estimator tends to overestimate the true value of p, and the normalization tends to underestimate it. For more details and experimental data supporting the use of this formula for estimates of p based on sample sizes from two to ten participants, see Lewis (2001a).

Adjusting the initial estimate of p. Because this is a new procedure, this section contains a detailed illustration of the steps used to adjust an initial estimate of p. To start with, organize the problem discovery data in a table (for example, Table 9) that shows which participants experienced which problems.

Table 9. Hypothetical results for a problem-discovery usability study

Participant	Prob 1	Prob 2	Prob 3	Prob 4	Prob 5	Prob 6	Prob 7	Prob 8	Count	Proportion
1	x		x		x		x	x	5	0.63
2	x	x			x		x		4	0.50
3	x		x	x	x				4	0.50
4	x	x				x			3	0.38
Count	4	2	2	1	3	1	2	1	16	
Proportion	1.00	0.50	0.50	0.25	0.75	0.25	0.50	0.25		0.50

With four participants and eight observed problems, there are 32 cells in the table. The total number of problem occurrences is 16, so the initial estimate of p (p_{est}) is .50 (16/32). Note that averaging the proportion of problem occurrence across participants or across problems also equals .50.

To apply the Good-Turing adjustment, count the number of problems that occurred with only one participant. In Table 9, this happened for three problems (Problems 4, 6, and 8) out of the eight unique problems listed in the table. Thus, the value of GT_{adj} is .375 (3/8), and the value of $p_{est}/(1+ GT_{adj})$ is .36 (.5/1.375).

To apply the normalization adjustment, start by computing $1/n$, which in Table 9 is .25 (1/4). The value of $(p_{est} – 1/n)(1 – 1/n)$ is .19 (.25*.75).

The average of the two adjustments produces p_{adj}, which in this example equals .28 ((.36+.19)/2). In this example, the adjusted estimate of p is almost half of the initial estimate.

Using the adjusted estimate of p. Once you have an appropriate (adjusted) estimate for p, you can use the formula $1-(1-p)^n$ (derivable both from the binomial probability formula, Lewis, 1982, 1994, and from the Poisson probability formula, Nielsen and Landauer, 1993) for various values of n from, say, 1 to 20, to generate the curve of diminishing returns expected as a function of sample size. It is possible to get even more sophisticated, taking into account the fixed and variable costs of the evaluation (especially the variable costs associated with the study of additional participants) to estimate when running an additional participant will result in costs that exceed the value of the additional problems discovered (Lewis, 1994).

The Monte Carlo experiments reported in Lewis (2001a) demonstrated that an effective strategy for planning the sample size for a usability study is first to establish a problem discovery goal (for example, 90% or 95%). Run the first two participants and, based on those results, calculate the adjusted value of p using the equation in [7]. This provides an early indication of the likely required sample size, which might estimate the final sample size exactly or, more likely, underestimate by one or two participants (but will provide an early estimate of the required sample size). Collect data from two more participants (for a total of four). Recalculate the adjusted estimate of p using the equation in [7] and project the required sample size

using $1-(1-p)^n$. The estimated sample size requirement based on data from four participants will generally be highly accurate, allowing accurate planning for the remainder of the study. Practitioners should do this even if they have calculated a preliminary estimate of the required sample size from an adjusted value for *p* obtained from a previous study.

Figure 4 shows the predicted discovery rates for problems of differing likelihoods of observation during a usability study. Several independent studies have verified that these types of predictions fit observed data very closely for both usability and heuristic evaluations (Lewis, 1994; Nielsen and Landauer, 1993; Nielsen and Molich, 1990; Virzi, 1990, 1992; Wright and Monk, 1991). Furthermore, the predictions work both for predicting the discovery of individual problems with a given probability of detection and for modeling the discovery of members of sets of problems with a given mean probability of detection (Lewis, 1994). For usability studies, the sample size is the number of participants. For heuristic evaluations, the sample size is the number of evaluators.

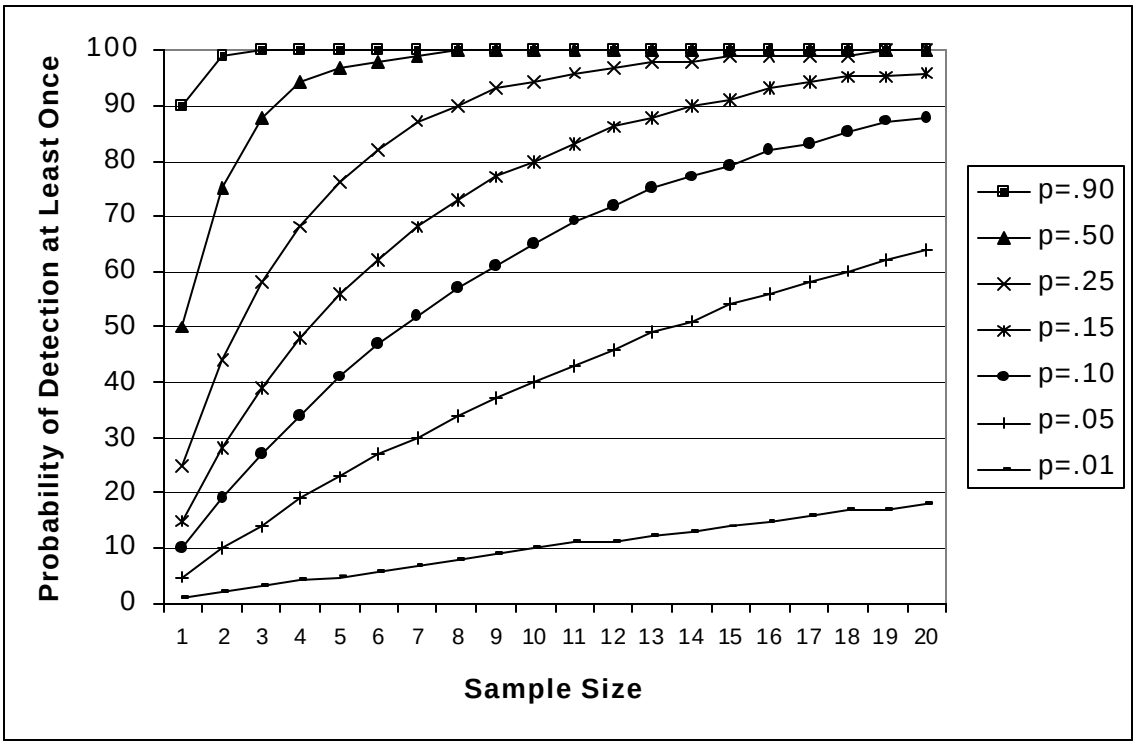

Figure 4. Predicted discovery as a function of problem likelihood

Table 10 (based on a table originally published in Lewis, 1994, but with updated computations to reduce the effect of round-off error) shows problem detection sample size requirements as a function of problem detection probability and the cumulative likelihood of detecting the problem at least once during the study. The required sample size for detecting the problem twice during a study appears in parentheses.

Table 10. Sample size requirements for problem discovery (formative) studies

Problem Occurrence Probability	**Cumulative Likelihood of Detecting the Problem at Least Once (Twice)**					
	0.50	**0.75**	**0.85**	**0.90**	**0.95**	**0.99**
0.01	69 (168)	138 (269)	189 (337)	230 (388)	299 (473)	459 (662)
0.05	14 (34)	28 (53)	37 (67)	45 (77)	59 (93)	90 (130)
0.10	7 (17)	14 (27)	19 (33)	22 (38)	29 (46)	44 (64)
0.15	5 (11)	9 (18)	12 (22)	15 (25)	19 (30)	29 (42)
0.25	3 (7)	5 (10)	7 (13)	9 (15)	11 (18)	17 (24)
0.50	1 (3)	2 (5)	3 (6)	4 (7)	5 (8)	7 (11)
0.90	1 (2)	1 (2)	1 (3)	1 (3)	2 (3)	2 (4)

To use this information to establish a usability sample size, you need to determine three things.

First, what is the average likelihood of problem detection probability (p)? This plays a role similar to the role of variance in the previous examples. If you don't know this value (from previous studies or a pilot study), then you need to decide on the lowest problem detection probability that you want to (or have the resources to) tackle. The smaller this number, the larger is the required sample size.

Second, you need to determine what proportion of the problems that exist at that level you need (or have the resources) to discover during the study (in other words, the cumulative likelihood of problem detection). The larger this number, the larger the required sample size.

Finally, you need to decide whether you are willing to take single occurrences of problems seriously or if problems must appear at least twice before receiving consideration. Requiring two occurrences results in a larger sample size.

For values of p or problem-discovery goals that are outside of tabled values, you can use the formula in [8] (derived algebraically from $Goal=1-(1-p)^n$) to directly compute the sample size required for a given problem discovery goal (taking single occurrences of problems seriously) and value of p.

$$[8]\ n = \log(1 - Goal)/\log(1 - p)$$

In the example from Table 9, the adjusted value of p was .28. Suppose the practitioner decided that the appropriate problem-discovery goal was to find 97% of the discoverable problems. The computed value of n is 10.6 (log(.03)/log(.72), or -1.522/-.143). The practitioner can either round the sample size up to 11 or adjust the problem-discovery goal down to 96.3% ($1-(1-.28)^{10}$).

Lewis (1994) created a return-on-investment (ROI) model to investigate appropriate cumulative problem detection goals. It turned out that the appropriate goal depended on the average problem detection probability in the evaluation – the same value that has a key role in determining the sample size. The model indicated that if the expected value of p was small (say, around 0.10), practitioners should plan to discover about 86% of the problems. If the expected value of p was larger (say, around .25 or .50), practitioners should plan to discover about 98% of the problems. For expected values of p between 0.10 and 0.25, practitioners should interpolate between 87 and 97% to determine an appropriate goal for the percentage of problems to discover.

The cost of an undiscovered problem had a strong effect on the magnitude of the maximum ROI, but contrary to expectation, it had a minor effect on sample size at maximum ROI (Lewis, 1994). Usability practitioners should be aware of these costs in their settings and their effect on ROI (Boehm, 1981), but these costs have relatively little effect on the appropriate sample size for a usability study.

In summary, there is compelling evidence that the law of diminishing returns, based on the cumulative binomial probability formula, applies to problem discovery studies. To use this formula to determine an appropriate sample size, practitioners must form an idea about the expected value of *p* (the average likelihood of problem detection) for the study and the percentage of problems that the study should uncover. Practitioners can use the ROI model from Lewis (1994) or their own ROI formulas to estimate an appropriate goal for the percentage of problems to discover and can examine data from their own or published usability studies to get an initial estimate of *p* (which published studies to date indicate can range at least from 0.16 to 0.60). With these two estimates, practitioners can use Table 10 (or, for computations outside of tabled values, the appropriate equations) to estimate appropriate sample sizes for their usability studies.

It is interesting to speculate that a new product that has not yet undergone any usability evaluation is likely to have a higher *p* than an established product that has gone through several development iterations (including usability testing). This suggests that it is easier (takes fewer participants) to improve a completely new product than to improve an existing product (as long as that existing product has benefited from previous usability evaluation). This is related to the idea that usability testing is a hill-climbing procedure, in which the results of a usability test are applied to a product to push its usability up the hill. The higher up the hill you go, the more difficult it becomes to go higher, because you have already weeded out the problems that were easy to find and fix.

Practitioners who wait to see a problem at least twice before giving it serious consideration can see from Table 10 the sample size implications of this strategy. Certainly, all other things being equal, it is more important to correct a problem that occurs frequently than one that occurs infrequently. However, it is unrealistic to assume that the frequency of detection of a problem is the only criterion to consider in the analysis of usability problems. The best strategy is to consider problem frequency and other problem data (such as severity and likelihood of use) simultaneously to determine which problems are most important to correct rather than establishing a cutoff rule such as "fix every problem that appears two or more times."

Note that in contrast to the results reported by Virzi (1992), the results reported by Lewis (1994) did not indicate any consistent relationship between problem frequency and impact (severity). It is possible that this difference was due to the different methods used to assess severity (judgment-driven in Virzi, 1992; data-driven in Lewis, 1994). Thus, the safest strategy is for practitioners to assume independence of frequency and impact until further research resolves the discrepancy between the outcomes of these studies.

It is important for practitioners to consider the risks as well as the gains when using small samples for usability studies. Although the diminishing returns for inclusion of additional participants strongly suggest that the most efficient approach is to run a small sample (especially if *p* is high, if the study will be iterative, and if undiscovered problems will not have dangerous or expensive outcomes), human factors engineers and other usability practitioners must not become complacent regarding the risk of failing to detect low-frequency but important problems.

One could argue that the true number of possible usability problems in any interface is essentially infinite, with an essentially infinite number of problems with non-zero probabilities that are extremely close to zero. For the purposes of determining sample size, the *p* we are really dealing with is the *p* that represents the number of discovered problems divided by the number of discoverable problems, where the definition of a discoverable problem is vague, but almost certainly constrained by details of the experimental setting, such as the studied scenarios and tasks and the skill of the observer(s). Despite this vagueness and some

recent criticism of the use of *p* to model problem-discovery (Caulton, 2001; Woolrych and Cockton, 2001), these techniques seem to work reasonably well in practice (Turner et al., in press).

Examples of sample size estimation for problem-discovery (formative) studies. This section contains several examples illustrating the use of Table 10 as an aid in selecting an appropriate sample size for a problem-discovery study.

A. Given the following problem discovery criteria:

- Detect problems with average probability of: 0.25
- Minimum number of detections required: 1
- Planned proportion to discover: 0.90

The appropriate sample size is 9 participants.

B. Given the same discovery criteria, except the practitioner requires problems to be detected twice before receiving serious attention:

- Detect problems with average probability of: 0.25
- Minimum number of detections required: 2
- Planned proportion to discover: 0.90

The appropriate sample size would be 15 participants.

C. Returning to requiring a single detection, but increasing the planned proportion to discover to .99:

- Detect problems with average probability of: 0.25
- Minimum number of detections required: 1
- Planned proportion to discover: 0.99

The appropriate sample size would be 17 participants.

D. Given the following extremely stringent discovery criteria:

- Detect problems with average probability of: 0.01
- Minimum number of detections required: 1
- Planned proportion to discover: 0.99

The required sample size would be 459 participants (an unrealistic requirement in most settings, implying unrealistic study goals).

Note that there is no requirement to run the entire planned sample through the usability study before reporting clear problems to development and getting those problems fixed before continuing. These required sample sizes are total sample sizes, not sample sizes per iteration. The following testing strategy promotes efficient iterative problem discovery studies and is similar to strategies published by a number of usability specialists (Bailey et al., 1992; Fu et al., 2002; Kantner and Rosenbaum, 1997; Jeffries and Desurvire, 1992; Macleod et al., 1997; Nielsen, 1993; Rosenbaum, 1989).

1. Start with an expert (heuristic) evaluation or one-participant pilot study to uncover the obvious problems. Correct as many of these problems as possible before starting the iterative cycles with Step 2. List all unresolved problems and carry them to Step 2.
2. Watch a small sample of participants (for example, three or four) use the system. Record all observed usability problems. Calculate an adjusted estimate of *p* based on these results and re-estimate the required sample size.

3. Redesign based on the problems discovered. Focus on fixing high frequency and high impact problems. Fix as many of the remaining problems as possible. Record any outstanding problems so they can remain open for all following iterations.
4. Continue iterating until you have reached your sample size goal (or must stop for any other reason, such as you ran out of time).
5. Record any outstanding problems remaining at the end of testing and carry them over to the next product for which they are applicable.

This strategy blends the benefits of large and small sample studies. During each iteration, you observe only three participants before redesigning the system. Therefore, you can quickly identify and correct the most frequent problems (which means you waste less time watching the next set of participants encounter problems that you already know about). With five iterations, for example, the total sample size would be 15 participants. With several iterations you will identify and correct many less frequent problems because you record and track the uncorrected problems through all iterations.

Note that using this sort of iterative procedure affects estimates of p as you go along. The value of p in the system you end with should generally be lower than the p you started with (as long as the process of fixing problems doesn't create as many other problems). For this reason, it's a good idea to recompute the adjusted value of p after each iteration.

Evaluating sample size effectiveness given fixed n. Suppose you know you only have time to run a limited number of participants, are willing to treat a single occurrence of a problem seriously, and want to determine what you can expect to get out of a problem-discovery study with that number of participants. If that number were six, for example, examination of Table 10 indicates:

- You are almost certain to detect problems that have a .90 likelihood of occurrence (it only takes two participants to have a 99% cumulative likelihood of seeing the problem at least once).
- You are almost certain (between 95 and 99% likely) to detect problems that have a .50 likelihood of occurrence (for this likelihood of occurrence, the required sample size at 95% is 5, and at 99% is 7).
- You've got a reasonable chance (about 80% likely) of detecting problems that have a .25 likelihood of occurrence (for this likelihood of occurrence, the required sample size at 75% is 5,and at 85% is 7).
- You have a little better than even odds of detecting problems that have a .15 likelihood of occurrence (the required sample size at 50% is 5).
- You have a little less than even odds of detecting problems that have a .10 likelihood of occurrence (the required sample size at 50% is 7).
- You are not likely to detect many of the problems that have a likelihood of occurrence of .05 or .01 (for these likelihoods of occurrence, the required sample size at 50% is 14 and 69 respectively).

This analysis illustrates that although a problem-discovery study with a sample size of six participants will typically not discover problems with very low likelihoods of occurrence, the study is almost certainly worth conducting.

Applying this procedure to a number of different sample sizes produces Table 11. The cells in Table 11 are the probability of having a problem with a specified occurrence probability happen at least once during a usability study with the given sample size.

Table 11. Likelihood of discovering problems of probability p at least once in a study with sample size n

Problem Occurrence Probability (p)	*Sample Size (*n*)*						
	3	**6**	**9**	**12**	**15**	**18**	**21**
.01	0.03	0.06	0.09	0.11	0.14	0.17	0.19
.05	0.14	0.26	0.37	0.46	0.54	0.60	0.66
.10	0.27	0.47	0.61	0.72	0.79	0.85	0.89
.15	0.39	0.62	0.77	0.86	0.91	0.95	0.97
.25	0.58	0.82	0.92	0.97	0.99	0.99	1.00
.50	0.88	0.98	1.00	1.00	1.00	1.00	1.00
.90	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Estimating the number of problems available for discovery. Another approach to assessing sample size effectiveness is to estimate the number of undiscovered problems. Returning to the situation illustrated in Table 9, the adjusted estimate of *p* is .28 with four participants and eight unique problems. The estimated proportion of problems discovered with those four participants is .73 $(1-(1-.28)^4)$. If eight problems are about 73% of the total number of problems available for discovery, then the total number of problems available for discovery (given the constraints of the testing situation) is about 11 (8/.73). Thus, there appear to be about three undiscovered problems. With an estimate of only three undiscovered problems, the sample size of four is approaching adequacy.

Contrast this with the MACERR study described in Lewis (2001a) that had an estimated value of *p* of .16 with fifteen participants and 145 unique problems. For this study, the estimated proportion of discovered problems at the end of the test was .927 $(1-(1-.16)^{15})$. The estimate of the total number of problems available for discovery was about 156 (145/.927). With about 11 problems remaining available for discovery, it might be wise to run a few more participants.

On the other hand, with an estimated 92.7% of problems available for discovery extracted from the problem discovery space defined by the test conditions, it might make more sense to make changes to the test conditions (in particular, to make reasonable changes to the tasks) to create additional opportunities for problem discovery. This is one of many areas in which practitioners need to exercise professional judgment, using the available tables and formulas to guide that judgment.

Some tips on managing p. Because *p* (the average likelihood of problem discovery) is such an important factor in sample size estimation for usability tests, it generally makes sense to attempt to manage it (although in some situations, such management is out of a practitioner's control). Here are some ways to increase *p*:

- Use highly-skilled observers for usability studies.
- Use multiple observers rather than a single observer (Hertzum and Jacobsen, 2003).
- Focus evaluation on new products with newly-designed interfaces rather than older, more refined interfaces.
- Study less-skilled participants in usability studies (as long as they are appropriate participants).
- Make the user sample as heterogeneous as possible, within the bounds of the population to which you plan to generalize the results.
- Make the task sample as heterogeneous as possible.
- Emphasize complex rather than simple tasks.

- For heuristic evaluations, use examiners with usability and application domain expertise (double experts, Nielsen, 1992).
- For heuristic evaluations, if you must make a tradeoff between having a single evaluator spend a lot of time examining an interface versus having more examiners spend less time each examining an interface, choose the latter option (Dumas, Sorce, and Virzi, 1995; Virzi, 1997).

Note that some of the tips for increasing p are the opposite of those that reduce measurement variability.

Sample Sizes for Non-Traditional Areas of Usability Evaluation

Non-traditional areas of usability evaluation include activities such as the evaluation of visual design and marketing materials. As with traditional areas of evaluation, the first step is to determine if the evaluation is comparative/parameter estimation or problem-discovery.

Part of the problem with non-traditional areas is that there is less information regarding the values of the variables needed to estimate sample sizes. Another issue is whether these areas are inherently focused on detecting more subtle effects than is the norm in usability testing, which has a focus on large low-variability effects (and correspondingly small sample size requirements). Determining this requires the involvement of someone with domain expertise in these non-traditional areas. It seems, however, that even these non-traditional areas would benefit from focusing on the discovery of large low-variability effects. Only if there was a business case that the investment in a study to detect small, highly-variable effects would ultimately pay for itself should you conduct such a study.

For example, in *The Survey Research Handbook*, Alreck and Settle (1985) point out that the reason that survey samples rarely contain fewer than several hundred respondents is due to the cost structure of surveys. The fixed costs of the survey include activities such as determining information requirements, identifying survey topics, selecting a data collection method, writing questions, choosing scales, composing the questionnaire, etc. For this type of research, the additional or 'marginal' cost of including hundreds of additional respondents can be very small relative to the fixed costs. Contrast this with the cost (or feasibility) of adding participants to a usability study in which there might be as little as a week or two between the availability of testable software and the deadline for affecting the product, with resources limiting the observation of participants to one at a time and the test scenarios requiring two days to complete. The potentially high cost of observing participants in usability tests is one reason why usability researchers have devoted considerable attention to sample size estimation, despite some assertions that sample size estimation is relatively unimportant (Wixon, 2003).

"Since the numbers don't know where they came from, they always behave just the same way, regardless." (Lord, 1953, p. 751) What potentially differs for non-traditional areas of usability evaluation isn't the behavior of numbers or statistical procedures, but the researchers' goals and economic realities.

Confidence Intervals

A major trend in modern statistical evaluation has been a reduced focus on hypothesis testing and a move toward more informative analyses such as effect sizes and confidence intervals (Landauer, 1997). For most applied usability work, confidence intervals are more useful than effect sizes because they have the same units of measurement as the variables from which they are computed. Even when confidence intervals are very wide, they can still be informative, so practitioners should routinely report confidence intervals for their measurements. Although 95% confidence is a commonly used level, confidence as low as 80% will often be appropriate for applied usability measurements (Nielsen, 1997).

Intervals Based on t-Scores

The formulas for the computation of confidence intervals based on t-scores are algebraically equivalent to those used to estimate required sample sizes for measurement-based usability tests, but isolate the critical difference (d) instead of the sample size (n), as shown in [9].

[9] $d = \text{sem} * t_{crit}$

where sem is the standard error of the mean (the standard deviation, *S*, divided by the square root of the sample size, *n*) and t_{crit} is the *t*-value associated with the desired level of confidence (found in a *t*-table, available in most statistics texts). (Practitioners who are concerned about departures from normality can perform a logarithmic transformation on their raw data before computing the confidence interval, then transform the data back to report the mean and confidence interval limits.)

For example, suppose that a task in a usability test with seven participants has an average completion time of 5.4 minutes with a standard deviation of 2.2 minutes. The sem is .83 ($2.2/(7^{1/2})$). For 90% confidence and 6 (*n* - 1) degrees of freedom, the tabled value of *t* is 1.943. The computed value of *d* is 1.6 (.83 * 1.943), so the 90% confidence interval is 5.4 ± 1.6 minutes.

As a second example, suppose that the results of a within-subjects test of the time required for two installation procedures showed that the mean of the difference scores (Version A minus Version B) was 2 minutes with a standard deviation of 2 minutes for a sample size of 8 participants. The sem is .71 ($2/(8^{1/2})$). For 95% confidence and 7 (*n* - 1) degrees of freedom, the tabled value of *t* is 2.365. The computed value of *d* is 1.7 (.71 * 2.365), so the 95% confidence interval is 2.0 ± 1.7 minutes (ranging from 0.3 to 3.7 minutes). Because the confidence interval does not contain 0, this interval indicates that with alpha of .05 (where alpha is 1 minus the confidence expressed as a proportion rather than a percentage) you should reject the null hypothesis of no difference. The evidence indicates that Version A takes longer than Version B. The major advantage of a confidence interval over a significance test is that you also know with 95% confidence that the magnitude of the difference is probably no less then 0.3 minutes and no greater than 3.7 minutes. If the versions are otherwise equal, then Version B is the clear winner. If the cost of Version B is greater than the cost of Version A (for example, due to a need to license a new technology for Version B), then the decision about which version to implement is more difficult, but is certainly aided by having an estimate of the upper and lower limits of the difference between the two versions.

Binomial Confidence Intervals

As discussed above, confidence intervals constructed around a mean can be very useful. Many usability measurements, however, are proportions or percentages computed from count data rather than means. For example, a usability defect rate for a specific problem is the proportion computed by dividing the number of participants who experience the problem divided by the total number of participants.

The statistical term for a study designed to estimate proportions is a *binomial experiment*, because a given problem either will or will not occur for each trial (participant) in the experiment. For example, a participant either will or will not install an option correctly. The point estimate of the defect rate is the observed proportion of failures (*p*). However, the likelihood is very small that the point estimate from a study is exactly the same as the true percentage of failures, especially if the sample size is small (Walpole, 1976). To compensate for this, you can calculate interval estimates that have a known likelihood of containing the true proportion (Steele and Torrie, 1960). You can use these binomial confidence intervals to describe the proportion of usability defects effectively, often with only a small sample (Lewis, 1996a). Cordes and Lentz (1986) and Lewis (1996a) provided BASIC programs for the computation of binomial confidence intervals. There are similar programs available at the website of the Southwest Oncology Group Statistical Center (SOGSC, 2004 – http://www.swogstat.org/stat/public/binomial_conf.htm), the GraphPad website (GraphPad, 2004 – http://graphpad.com/quickcalcs/ConfInterval1.cfm), and the Measuring Usability website (Sauro, 2004 – http://www.measuringusability.com/conf_intervals.htm).

Some programs (Cordes and Lentz, 1986; Lewis, 1996a; SOGSC, 2004) produce binomial confidence intervals that always contain the exact binomial confidence interval. Other programs (GraphPad, 2004; Sauro, 2004) also produce a new type of interval called *approximate* binomial confidence intervals (Agresti and Coull, 1998). Exact and approximate binomial confidence intervals differ in a number of ways. An exact binomial confidence interval guarantees that the actual confidence is equal to or greater than the nominal

confidence. An approximate interval guarantees that the average of the actual confidence in the long run will be equal to the nominal confidence, but for any specific test, the actual confidence could be lower than the nominal confidence. On the other hand, approximate binomial confidence intervals tend to be narrower than exact intervals. When sample sizes are large ($n > 100$), the two types of intervals are virtually indistinguishable. When sample sizes are small, though, there can be a considerable difference in the width of the intervals, especially when the observed proportion is close to 0 or 1. The exact interval often has an actual confidence closer to 99% when the nominal confidence is 95%, making it too conservative.

Monte Carlo studies that have compared exact and approximate binomial confidence intervals using standard statistical distributions (Agresti and Coull, 1998) and data from usability studies (Sauro and Lewis, 2005) generally support the use of approximate rather than exact binomial confidence intervals. When the actual confidence of an approximate binomial confidence interval is below the nominal level, the actual level tends to be close to the nominal level (for example, Agresti and Coull, 1998, found that the actual level for 95% approximate binomial confidence intervals using the adjusted-Wald method was never less than 89%). "In forming a 95% confidence interval, is it better to use an approach that guarantees that the actual coverage probabilities are *at least* .95 yet typically achieves coverage probabilities of about .98 or.99, or an approach giving narrower intervals for which the actual coverage probability could be less than .95 but is usually quite *close* to .95? For most applications, we would prefer the latter" (Agresti and Coull, 1998, p. 125). This conclusion, that using approximate binomial confidence intervals will tend to produce superior decisions relative to the use of exact intervals, seems to apply to usability test data (Sauro and Lewis, 2005). If, however, it is critical for a specific test to achieve or exceed the nominal level of confidence, then it is reasonable to use an exact binomial confidence interval.

When using binomial confidence intervals, note that if the failure rate is fairly high, you do not need a very large sample to acquire convincing evidence of failure. In the first evaluation of a wordless graphic instruction (Lewis and Pallo, 1991), 9 of 11 installations (82%) were incorrect. The exact 90% binomial confidence interval for this outcome ranged from .53 to .97. This interval allowed us to argue that without intervention, the failure rate for installation would be at least 53% (and more likely closer to the observed 82%).

This suggests that a reasonable strategy for binomial experiments is to start with a small sample size and record the number of failures. From these results, compute a confidence interval. If the lower limit of the confidence interval indicates an unacceptably high failure rate, stop testing. Otherwise, continue testing and evaluating in increments until you reach a specified level of precision or you reach the maximum sample size allowed for the study.

This method can rapidly demonstrate with a small sample that a usability defect is unacceptably high if the criterion is low and the true defect rate is high. Although the confidence interval will be wide (50 percentage points in the graphic symbols example), the lower limit of the interval may be clearly unacceptable. When the true defect rate is low or the criterion is high, this procedure may not work without a large sample size. The decision to continue sampling or to stop the study should be determined by a reasonable business case that balances the cost of continued data collection against the potential cost of allowing defects to go uncorrected.

You cannot use this procedure with small samples to prove that a success rate is acceptably high. With small samples, even if the observed defect percentage is 0 or close to 0%, the interval will be wide, so it will probably include defect percentages that are unacceptable. For example, suppose you have run five participants through a task, and all five have completed the task successfully. The 90% confidence interval on the percentage of defects for these results ranges from 0 to 45%, with a 45% defect rate almost certainly unacceptable. If you had fifty out of fifty successful task completions, the 90% binomial confidence interval would range from 0 to 6%, which would indicate a greater likelihood of the true defect rate being close to 0%. The moral of the story is that it is relatively easy to prove (requires a small sample) that a product is unacceptable, but it is difficult to prove (requires a large sample) that a product is acceptable.

Standardized Usability Questionnaires

Standardized satisfaction measures offer many advantages to the usability practitioner. Specifically, standardized measurements provide objectivity, replicability, quantification, economy, communication, and scientific generalization (Nunnally, 1978). The first published standardized usability questionnaires appeared in the late 1980s (Chin et al., 1988; Kirakowski and Dillon, 1988). Questionnaires focused on the measurement of computer satisfaction preceded these questionnaires (for example, the Gallagher Value of MIS Reports Scale and the Hatcher and Diebert Computer Acceptance Scale – see LaLomia and Sidowski, 1990, for a review), but these questionnaires were not applicable to scenario-based usability tests.

The most widely used standardized usability questionnaires are the Questionnaire for User Interaction Satisfaction (QUIS, Chin et al., 1988), the Software Usability Measurement Inventory (SUMI, Kirakowski, 1996; Kirakowski and Corbett, 1993), the Post-Study System Usability Questionnaire (PSSUQ, Lewis, 1992, 1995, 2002), and the Software Usability Scale (SUS, Brooke, 1996). The most common application of these questionnaires is at the end of a test (after completing a series of test scenarios). The After-Scenario Questionnaire (ASQ, Lewis, 1991b) is a short three-item questionnaire designed for administration immediately following the completion of a test scenario. The ASQ takes less than a minute to complete. The longer standard questionnaires typically have completion times of less than ten minutes (Dumas, 2003).

The primary measures of standardized questionnaire quality are reliability (consistency of measurement) and validity (measurement of the intended attribute) (Nunnally, 1978). There are several ways to assess reliability, including test-retest and split-half reliability. The most common method for the assessment of reliability is coefficient alpha, a measurement of internal consistency. Coefficient alpha can range from 0 (no reliability) to 1 (perfect reliability). Measures that can affect an individual's future, such as IQ tests or college entrance exams should have a minimum reliability of .90 (preferably, reliability greater than .95). For other research or evaluation, measurement reliability in the range of .70 to .80 is acceptable (Landauer, 1997; Nunnally, 1978).

A questionnaire's validity is the extent to which it measures what it claims to measure. Researchers commonly use the Pearson correlation coefficient to assess criterion-related validity (the relationship between the measure of interest and a different concurrent or predictive measure). These correlations do not have to be large to provide evidence of validity. For example, personnel selection instruments with validities as low as .30 or .40 can be large enough to justify their use (Nunnally, 1978). Another approach to validity is content validity, typically assessed through the use of factor analysis (which also helps questionnaire developers discover or confirm clusters of related items that can form reasonable subscales).

Regarding the appropriate number of scale steps, it is true that more scale steps are better than fewer scale steps, but with rapidly diminishing returns. The reliability of individual items is a monotonically increasing function of the number of steps (Nunnally, 1978). As the number of scale steps increase from 2 to 20, the increase in reliability is very rapid at first, but tends to level off at about 7. After 11 steps there is little gain in reliability from increasing the number. The number of steps in an item is very important for measurements based on a single item, but is less important when computing measurements over a number of items (as in the computation of an overall or subscale score).

The QUIS

The QUIS (Chin et al., 1988; Shneiderman, 1987, see http://lap.umd.edu/QUIS/) is a product of the Human-Computer Interaction Lab at the University of Maryland. Its use requires the purchase of a license. Chin et al. (1988) evaluated several early versions of the QUIS (Versions 3 through 5). They reported an overall reliability (coefficient alpha) of .94, but did not report any subscale reliability.

The QUIS is currently at Version 7. This version includes demographic questions, an overall measure of system satisfaction, and 11 specific interface factors. The QUIS is available in two lengths, short

(26 items) and long (71 items). The items are nine-point scales anchored with opposing adjective phrases (such as "confusing" and "clear" for the item, "Messages which appear on screen").

The CUSI and SUMI

The Human Factors Research Group (HFRG) at University College Cork published their first standardized questionnaire, the Computer Usability Satisfaction Inventory (CUSI), in 1988 (Kirakowski and Dillon). The CUSI was a 22-item questionnaire containing two subscales: Affect and Competence. Its overall reliability was .94, with .91 for Affect and .89 for Competence.

The HRFG replaced the CUSI with the SUMI (Kirakowski, 1996; Kirakowski and Corbett, 1993; see http://www.ucc.ie/hfrg/questionnaires/sumi//index.html), a questionnaire that has six subscales: Global, Efficiency, Affect, Helpfulness, Control, and Learnability. Its 50 items are statements (such as "The instructions and prompts are helpful.") to which participants indicate that they agree, are undecided, or disagree. The SUMI has undergone a significant amount of psychometric development and evaluation to arrive at its current form. The results of several sensitivity studies that had significant main effects of system, SUMI scales, and their interaction (for example, McSweeney, 1992, and Wiethoff et al., 1992) support its validity.

The reported reliabilities of the six subscales (measured with coefficient alpha) are:

- Global: .92
- Efficiency: .81
- Affect: .85
- Helpfulness: .83
- Control: .71
- Learnability: .82

One of the greatest strengths of the SUMI is the database of results that is available for the construction of interpretive norms. This makes it possible for practitioners to compare their results with those of similar products (as long as there are similar products in the database). Another strength is that the SUMI is available in different languages (such as UK English, American English, Italian, Spanish, French, German, Dutch, Greek, and Swedish). Like the QUIS, practitioners planning to use SUMI must purchase a license for its use (which includes questionnaires and scoring software). For an additional fee, a trained psychometrician at the HRFG will score the results and produce a report.

The SUS

Usability practitioners at Digital Equipment Corporation (DEC) developed the SUS in the mid 1980s (Dumas, 2003). The ten 5-point items of the SUS provide a unidimensional (no subscales) usability measurement that ranges from 0 to 100. In the first published account of the SUS, Brooke (1996) stated that the SUS was robust, reliable, and valid, but did not publish the specific reliability or validity measurements. With regard to validity, "it correlates well with other subjective measures of usability (e.g., the general usability subscale of the SUMI)" (Brooke, 1996, p. 194). DEC has copyrighted the SUS, but according to Brooke (1996), "the only prerequisite for its use is that any published report should acknowledge the source of the measure" (p. 194).

The PSSUQ and CSUQ

The PSSUQ is a questionnaire designed for the purpose of assessing users' perceived satisfaction with their computer systems. It has its origin in an internal IBM project called SUMS (System Usability MetricS), headed by Suzanne Henry in the late 1980s. A team of human factors engineers and usability specialists working on SUMS created a pool of 7-point scale items based on the work of Whiteside et al. (1988), and from that pool selected 18 items to use in the first version of the PSSUQ (Lewis, 1992). Each item was positively worded, with the scale anchors "Strongly Agree" at the first scale position (1) and "Strongly Disagree" at the last scale position (7). A Not Applicable (NA) choice and a comment area were available for each item (see Lewis, 1995 for examples of the appearance of the items).

The development of the Computer System Usability Questionnaire (CSUQ) followed the development of the first version of the PSSUQ. Its items are identical to those of the PSSUQ except that their wording is appropriate for use in field settings or surveys rather than in a scenario-based usability test, making it, essentially, an alternate form of the PSSUQ. For a discussion of CSUQ research and comparison of the PSSUQ and CSUQ items, see Lewis (1995).

An unrelated series of IBM investigations into customer perception of usability revealed a common set of five usability characteristics associated with usability by several different user groups (Doug Antonelli, personal communication, January 5, 1991). The 18-item version of the PSSUQ addressed four of these five characteristics (quick completion of work, ease of learning, high-quality documentation and online information, and functional adequacy), but did not address the fifth (rapid acquisition of productivity). The second version of the PSSUQ (Lewis, 1995) included an additional item to address this characteristic, bringing the total number of items up to 19.

Lewis (2002) conducted a psychometric evaluation of the PSSUQ using data from several years of usability studies (primarily studies of speech dictation systems, but including studies of other types of applications). The results of a factor analysis on these data were consistent with earlier factor analyses (Lewis, 1992, 1995) used to define three PSSUQ subscales: System Usefulness (SysUse), Information Quality (InfoQual), and Interface Quality (IntQual). Estimates of reliability were also consistent with those of earlier studies. Analyses of variance indicated that variables such as the specific study, developer, state of development, type of product, and type of evaluation significantly affected PSSUQ scores. Other variables, such as gender and completeness of responses to the questionnaire, did not. Norms derived from the new data correlated strongly with norms derived from earlier studies.

Significant correlation analyses indicated scale validity (Lewis, 1995). For a sample of 22 participants who completed all PSSUQ and ASQ items in a usability study (Lewis et al., 1990), the overall PSSUQ score correlated highly with the sum of the ASQ ratings that participants gave after completing each scenario ($r(20) = .80$, $p = .0001$). The overall PSSUQ score correlated significantly with the percentage of successful scenario completions ($r(29) = -.40$, $p = .026$). SysUse ($r(36) = -.40$, $p = .006$) and IntQual ($r(35) = -.29$, $p = .08$) also correlated with the percentage of successful scenario completions.

One potential criticism of the PSSUQ has been that some items seemed redundant, and that this redundancy might inflate estimates of reliability. Lewis (2002) investigated the effect of removing three items from the second version of the PSSUQ (Items 3, 5, and 13). With these items removed, the reliability of the overall PSSUQ score (using coefficient alpha) was .94 (remaining very high), and the reliabilities of the three subscales were:

- SysUse: .90
- InfoQual: .91
- IntQual: .83

All of the reliabilities exceeded .80, indicating sufficient reliability to be valuable as usability measurements (Anastasi, 1976; Landauer, 1997). Thus, the third (and current) version of the PSSUQ has 16 7-point scale items (see Table 12 for the items and their normative scores from Lewis, 2002).

Note that the scale construction is such that lower scores are better than higher scores, and that the means of the items and scales all fall below the scale midpoint of 4. With the exception of Item 7 ("The system gave error messages that clearly told me how to fix problems."), the upper limits of the confidence intervals are below 4. This shows that practitioners should not exclusively use the scale midpoint as a reference from which they would judge participants' perceptions of usability. Rather, they should also use the norms shown in Table 12 (and comparison with these norms is probably more meaningful than comparison with the scale midpoint).

Table 12. PSSUQ Version 3 items, scales, and normative scores (99% confidence intervals)

Item/ Scale	Item Text/Scale Scoring Rule	Norms (99% CI) Lower Limit	Mean	Upper Limit
Q1	Overall, I am satisfied with how easy it is to use this system	2.60	**2.85**	3.09
Q2	It was simple to use this system.	2.45	**2.69**	2.93
Q3	I was able to complete the tasks and scenarios quickly using this system.	2.86	**3.16**	3.45
Q4	I felt comfortable using this system.	2.40	**2.66**	2.91
Q5	It was easy to learn to use this system.	2.07	**2.27**	2.48
Q6	I believe I could become productive quickly using this system.	2.54	**2.86**	3.17
Q7	The system gave error messages that clearly told me how to fix problems.	3.36	**3.70**	4.05
Q8	Whenever I made a mistake using the system, I could recover easily and quickly.	2.93	**3.21**	3.49
Q9	The information (such as on-line help, on-screen messages and other documentation) provided with this system was clear.	2.65	**2.96**	3.27
Q10	It was easy to find the information I needed.	2.79	**3.09**	3.38
Q11	The information was effective in helping me complete the tasks and scenarios.	2.46	**2.74**	3.01
Q12	The organization of information on the system screens was clear. Note: The "interface" includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the microphone, and the screens (including their graphics and language).	2.41	**2.66**	2.92
Q13	The interface of this system was pleasant.	2.06	**2.28**	2.49
Q14	I liked using the interface of this system.	2.18	**2.42**	2.66
Q15	This system has all the functions and capabilities I expect it to have.	2.51	**2.79**	3.07
Q16	Overall, I am satisfied with this system.	2.55	**2.82**	3.09
SysUse	Average Items 1 through 6.	2.57	**2.80**	3.02
InfoQual	Average Items 7 through 12.	2.79	**3.02**	3.24
IntQual	Average Items 13 through 15.	2.28	**2.49**	2.71
Overall	Average Items 1 through 16.	2.62	**2.82**	3.02

Table notes: SysUse = system usefulness, InfoQual = information quality, IntQual = interface quality, CI = confidence interval. Means appear in bold face. Scores can range from 1 (strongly agree) to 7 (strongly disagree), with lower scores better than higher scores.

The way that Item 7 stands out from the others indicates:

- It should not surprise practitioners if they find this in their own data.
- It is a difficult task to provide usable error messages throughout a product.
- It may well be worth the effort to focus on providing usable error messages.
- If practitioners find the mean for this item to be equal to or less than the mean of the other items in InfoQual (assuming they are in line with the norms), they have been successful in creating better-than-average error messages.

The consistent pattern of relatively poor ratings for InfoQual versus IntQual (seen across all of the studies – for details and complete normative data, see Lewis, 2002) suggests that practitioners who find this pattern in their data should not conclude that they have poor documentation or a great interface. Suppose, however, that this pattern appeared in the first iteration of a usability evaluation and the developers decided to emphasize improvement to the quality of their information. Any subsequent decline in the difference between InfoQual and IntQual would be evidence of a successful intervention.

Another potential criticism of the PSSUQ is that the items do not follow the typical convention of varying the tone of the items so that half of the items elicit agreement and the other half elicit disagreement. The rationale for the decision to consistently align the items was to make it as easy as possible for participants to complete the questionnaire. With consistent item alignment, the proper way to mark responses on the items is clearer, potentially reducing response errors due to participant confusion. Also, the use of negatively worded items can produce a number of undesirable effects (Barnette, 2000; Ibrahim, 2001), including problems with internal consistency and factor structure. The setting in which balancing the tone of the items is likely to be of greatest value is when participants do not have a high degree of motivation for providing reasonable and honest responses (for example, in clinical and educational settings). Obtaining reasonable and honest responses is rarely a problem in most usability testing settings.

Additional key findings and conclusions from Lewis (2002) were:

- There was no evidence of response styles (especially, no evidence of extreme response style) in the PSSUQ data.
- Because there is a possibility of extreme response and acquiescence response styles in cross-cultural research (Baumgartner and Steenkamp, 2001; Clarke, 2001; Grimm and Church, 1999, van de Vijver and Leung, 2001), practitioners should avoid using questionnaires for cross-cultural comparison unless that use has been validated. Other types of group comparisons with the PSSUQ are valid because any effect of response style should cancel out across experimental conditions.
- Scale scores from incomplete PSSUQs were indistinguishable from those computed from complete PSSUQs. These data do not provide information concerning how many items a participant might ignore and still produce reliable scale scores. They do suggest that, in practice, participants typically complete enough items to produce reliable scale scores.

The similarity of psychometric properties across the various versions of the PSSUQ, despite the passage of time and differences in the types of systems studied, provides evidence of significant generalizability for the questionnaire, supporting its use by practitioners for measuring participant satisfaction with the usability of tested systems. Due to its generalizability, practitioners can confidently use the PSSUQ when evaluating different types of products and at different times during the development process. The PSSUQ can be especially useful in competitive evaluations (for an example, see Lewis, 1996b) or when tracking changes in usability as a function of design changes made during development. Practitioners and researchers are free to use the PSSUQ and CSUQ (no license fees), but anyone using them should cite the source.

The ASQ

The ASQ (Lewis, 1991b, 1995) is an extremely short questionnaire (three 7-point scale items using the same format as the PSSUQ). The items address three important aspects of user satisfaction with system usability: ease of task completion ("Overall, I am satisfied with the ease of completing the tasks in this scenario."), time to complete a task ("Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario."), and adequacy of support information ("Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing tasks.") The overall ASQ score is the average of responses to these three items.

Because the questionnaire is short, it takes very little time for participants to complete, an important practical consideration for usability studies. Measurements of ASQ reliability (using coefficient alpha) have ranged from .90 to .96 (Lewis, 1995). A significant correlation between ASQ scores and successful scenario completion ($r(46) = -.40$, $p < .01$) in Lewis et al. (1990, analysis reported in Lewis, 1995) provided evidence of concurrent validity. Like the PSSUQ and CSUQ, the ASQ is available for free use by practitioners and researchers, but anyone using the ASQ should cite the source.

WRAPPING UP

Getting More Information about Usability Testing

This chapter has provided fundamental and some advanced information about usability testing, but there is only so much that you can cover in a single chapter. For additional chapter-length treatments of the basics of usability testing, see Nielsen (1997) and Dumas (2003). There are also two well-known books devoted to the topic of usability testing.

Dumas and Redish (1999) is one of these book-length treatments of usability testing. The content and references are somewhat dated. The 1999 copyright date is a bit misleading, as the body of the book has not changed since its 1993 edition. The 1999 edition does include a new preface and some updated reading recommendations, and provides an excellent coverage of the fundamentals of usability testing.

The other well-known usability testing book is Rubin (1994). Like Dumas and Redish (1999), the content and references are ten years out of date. It, too, covers the fundamentals of usability testing (which haven't really changed for 20 years) very well and contains many useful samples of a variety of testing-related forms and documents.

For late-breaking developments in usability research and practice, there are a number of annual conferences that have usability evaluation as a significant portion of their content. Companies making a sincere effort in the professional development of their usability practitioners should ensure that their personnel have access to the proceedings of these conferences and should support attendance at one or more of these conferences at least every few years. These major conferences are:

- Usability Professionals Association (see http://www.upassoc.org/)
- Human-Computer Interaction International (see http://www.hci-international.org/)
- ACM Special Interest Group in Computer-Human Interaction (http://www.acm.org/sigchi/)
- Human Factors and Ergonomics Society (see http://hfes.org/)
- INTERACT (held every two years, for example, see http://www.interact2005.org/)

A Research Challenge: Improved Understanding of Usability Problem Detection

The recently published papers that have questioned the reliability of usability problem discovery (Hertzum and Jacobsen, 2003; Kessner et al., 2001; Molich et al., 1998, 2004) have raised a number of questions. Dumas (2003, p. 1112) responded, "It is not clear why there is so little overlap in problems. Are slight variations in method the cause? Are the problems really the same but just described differently? We look to further research to sort out the possibilities." Hertzum and Jacobsen (2003) noted the following as potential causes of lack of reliability across evaluations:

- Vague goal analyses that lead to variability in task scenarios
- Vague evaluation procedures
- Vague problem criteria that lead to acceptance of anything as a usability problem

Developing a better understanding of why these studies produced their results, which are so at odds with the apparent success of usability testing (Al-Awar et al., 1981; Bailey, 1993; Bailey et al., 1992; Gould et al., 1987; Kelley, 1984; Kennedy, 1982; Lewis, 1982; Lewis, 1996b; Marshall et al., 1990; Ruthford and Ramey, 2000), should be one of the top usability research efforts of the coming decade. An improved understanding might provide guidance about how or whether practitioners should change the way they conduct usability tests. One of the most important components of this research effort should be to investigate the cognitive mechanisms that underlie the detection of usability problems. Even after over 20 years of professional practice with usability methods, there is still no general consensus on the boundaries of what constitutes a usability problem, or on the appropriate levels of description of usability problems

(Lewis, 2001b). Doctoral candidates should take note – this is a topic rich with theoretical and practical consequences!

For example, it seems reasonable that the task of the observer in a usability study involves classical signal-detection issues (Massaro, 1975). The observer monitors participant behavior, and at any given moment must decide whether that observed behavior is indicative of a usability problem. Thus, there are two ways for an observer to make correct decisions (rejecting non-problem behaviors correctly, identifying problem behaviors correctly) and two ways to make incorrect decisions (identifying non-problem behaviors as indicative of a usability problem, failing to identify problem behaviors as indicative of a usability problem). In signal detection terms, the names for these right and wrong decisions are Correct Rejection, Hit, False Alarm, and Miss. The rates for these types of decisions depend independently on both the skill and the bias of the observer. Applying signal detection theory to the assessment of the skill and bias of usability test observers is a potentially rich, but to date untapped, area of research, with potential application for both selection and training of observers.

Usability Testing: Yesterday, Today, and Tomorrow

It seems clear that usability testing (both summative and formative) is here to stay, and that its general form will remain similar to the forms that emerged in the late 1970s and early 1980s. The last 25 years have seen the introduction of more usability evaluation techniques and some consensus (and some continuing debate) on the conditions under which to use the various techniques (of which usability testing is a major one). In the last 15 years, usability researchers have made significant progress in the areas of standardized usability questionnaires and sample size estimation for formative usability tests. As we look to the future, usability practitioners should monitor the continuing research that will almost certainly take place in developing a better understanding of the cognitive mechanisms of usability problem discovery because such an understanding has the potential to increase the reliability of usability testing.

In the mean time, practitioners will continue to perform usability tests, exercising professional judgment as required. Usability testing is not a perfect usability evaluation method in the sense that it does not guarantee the discovery of all possible usability problems, but it doesn't have to be perfect to be useful and effective. It is, however, important to understand its strengths, limitations, and current best practices to ensure its proper (most effective) use.

Acknowledgements

I want to thank Gavriel Salvendy for giving me the opportunity and encouragement to write this chapter. I also want to express my deepest appreciation to the colleagues who took the time to review the first draft under a very tight deadline – Patrick Commarford, Richard Cordes, Barbara Millet, Jeff Sauro, and Wallace Sadowski.

REFERENCES

Aaker, D. A., and Day, G. S. (1986). *Marketing research*. New York, NY: John Wiley.

Abelson, R. P. (1995). *Statistics as principled argument*. Mahwah, NJ: Lawrence Erlbaum.

Al-Awar, J., Chapanis, A., and Ford, R. (1981). Tutorials for the first-time computer user. *IEEE Transactions on Professional Communication*, *24*, 30-37.

Alreck, P. L., and Settle, R. B. (1985). *The survey research handbook*. Homewood, IL: Richard D. Irwin, Inc.

Alty, J. L. (1992). Can we measure usability? In *Proceedings of Advanced Information Systems* (pp. 95-106). London, UK: Learned Information.

Anastasi, A. (1976). *Psychological testing*. New York, NY: Macmillan.

Andre, T. S., Belz, S. M, McCreary, F. A., and Hartson, H. R. (2000). Testing a framework for reliable classification of usability problems. In *Proceedings of the IEA 2000/HFES 2000 Congress* (pp. 573-576). Santa Monica: CA: Human Factors and Ergonomics Society.

ANSI. (2001). *Common industry format for usability test reports* (ANSI-NCITS 354-2001). Washington, DC: American National Standards Institute.

Aykin, N. M., and Aykin, T. (1991). Individual differences in human-computer interaction. *Computers and Industrial Engineering*, *20*, 373-379.

Bailey, G. (1993). Iterative methodology and designer training in human-computer interface design. In *INTERCHI '93 Conference Proceedings* (pp. 198-205). New York, NY: ACM.

Bailey, R. W., Allan, R. W., and Raiello, P. (1992). Usability testing vs. heuristic evaluation: A head to head comparison. In *Proceedings of the Human Factors and Ergonomics Society 36th Annual Meeting* (pp. 409-413). Atlanta, GA: Human Factors and Ergonomics Society.

Banks, S. (1965). *Experimentation in marketing*. New York, NY: McGraw-Hill.

Barnette, J. J. (2000). Effects of stem and Likert response option reversals on survey internal consistency: If you feel the need, there is a better alternative to using those negatively worded stems. *Educational and Psychological Measurement*, *60*, 361-370.

Baumgartner, H., and Steenkamp, J. B. E. M. (2001). Response styles in marketing research: A cross-national investigation. *Journal of Marketing Research*, *38*, 143-156.

Bennett, J. L. (1979). The commercial impact of usability in interactive systems. *Infotech State of the Art Report: Man/Computer Communication*, *2*, 289-297.

Berry, D. C., and Broadbent, D. E. (1990). The role of instruction and verbalization in improving performance on complex search tasks. *Behaviour & Information Technology*, *9*, 175-190.

Bevan, N., Kirakowski, J., and Maissel, J. (1991). What is usability? In H. J. Bullinger (Ed.), *Human Aspects in Computing, Design and Use of Interactive Systems and Work with Terminals, Proceedings of the Fourth International Conference on Human Computer Interaction* (pp. 651-655). Stuttgart, Germany: Elsevier Science Publishers.

Bias, R. G., and Mayhew, D. J. (1994). *Cost-justifying usability*. Boston, MA: Academic Press.

Blalock, H. M. (1972). *Social statistics*. New York, NY: McGraw-Hill.

Boehm, B. W. (1981). *Software engineering economics*. Englewood Cliffs, NJ: Prentice-Hall.

Boren, T., and Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. *IEEE Transactions on Professional Communications, 43*, 261-278.

Bowers, V., and Snyder, H. (1990). Concurrent versus retrospective verbal protocols for comparing window usability. In *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 1270-1274). Santa Monica, CA: Human Factors Society.

Bradley, J. V. (1976). *Probability; decision; statistics*. Englewood Cliffs, NJ: Prentice-Hall.

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. Jordan, B. Thomas, and B. Weerdmeester (Eds.), *Usability Evaluation in Industry* (pp. 189-194). London, UK: Taylor and Francis.

Brown, F. E. (1980). *Marketing research: A structure for decision making*. Reading, MA: Addison-Wesley.

Campbell, D. T., and Stanley, J. C. (1963). *Experimental and quasi-experimental designs for research*. Chicago, IL: Rand McNally.

Caulton, D.A. (2001). Relaxing the homogeneity assumption in usability testing. *Behaviour & Information Technology, 20*, 1-7.

Chapanis, A. (1981). *Evaluating ease of use*. Unpublished manuscript prepared for IBM, available from J. R. Lewis.

Chapanis, A. (1988). Some generalizations about generalization. *Human Factors, 30*, 253-267.

Chin, J. P., Diehl, V. A., and Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In E. Soloway, D. Frye, and S. B. Sheppard (Eds.), *CHI '88 Conference Proceedings: Human Factors in Computing Systems* (pp. 213-218). Washington, D.C.: ACM.

Churchill, Jr., G. A. (1991). *Marketing research: Methodological foundations.* Ft. Worth, TX: Dryden Press.

Clarke, I. (2001). Extreme response style in cross-cultural research. *International Marketing Review, 18*, 301-324.

Cliff, N. (1987). *Analyzing multivariate data*. San Diego: CA: Harcourt Brace Jovanovich.

Cockton, G., and Lavery, D. (1999). A framework for usability problem extraction. In *Human Computer Interaction INTERACT '99* (pp. 344-352). Amsterdam, Netherlands: IOS Press.

Cordes, R. E. (1993). The effects of running fewer subjects on time-on-task measures. *International Journal of Human-Computer Interaction, 5*, 393-403.

Cordes, R. E. (2001). Task-selection bias: A case for user-defined tasks. *International Journal of Human-Computer Interaction, 13*, 411-419.

Cordes, R. E., and Lentz, J. L. (1986). *Usability-claims evaluation using a binomial test* (Tech. Report 82.0267). Tucson, AZ: International Business Machines Corp.

Desurvire, H. W., Kondziela, J. M., and Atwood, M. E. (1992). What is gained and lost when using evaluation methods other than empirical testing. In A. Monk, D. Diaper, and M. D. Harrison (Eds.), *Proceedings of HCI '92: People and Computers VII* (pp. 89-102). York, UK: Cambridge University Press.

Diamond, W. J. (1981). *Practical experiment designs for engineers and scientists*. Belmont, CA: Lifetime Learning Publications.

Dickens, J. (1987). The fresh cream cakes market: The use of qualitative research as part of a consumer research programme. In U. Bradley (ed.), *Applied Marketing and Social Research* (pp. 23-68). New York, NY: John Wiley.

Dumas, J. S. (2003). User-based evaluations. In J. A. Jacko and A. Sears (Eds.), *The Human-Computer Interaction Handbook* (pp. 1093-1117). Mahwah, NJ: Lawrence Erlbaum.

Dumas, J., and Redish, J. C. (1999). *A practical guide to usability testing*. Portland, OR: Intellect.

Dumas, J., Sorce, J., and Virzi, R. (1995). *Expert reviews: How many experts is enough?* In Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting (pp. 228-232). Santa Monica, CA: Human Factors and Ergonomics Society.

Ericsson, K. A., and Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, *87*, 215-251.

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, *35*, 379-383.

Fisher, J. (1991). Defining the novice user. *Behaviour & Information Technology*, *10*, 437-441.

Fowler, C. J. H., Macaulay, L. A., and Fowler, J. F. (1985). The relationship between cognitive style and dialogue style: an exploratory study. In P. Johnson and S. Cook (eds.), *People and Computers: Designing the Interface* (pp. 186-198). Cambridge, UK: Cambridge University Press.

Fu, L., Salvendy, G., and Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, *21*, 137-143.

Gawron, V. J., Drury, C. G., Czaja, S. J., and Wilkins, D. M. (1989). A taxonomy of independent variables affecting human performance. *International Journal of Man-Machine* Studies, *31*, 643-672.

Gordon, W., and Langmaid, R. (1988). Qualitative market research: A practitioner's and buyer's guide. Aldershot, UK: Gower.

Gould, J. D. (1988). How to design usable systems. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 757-789). Amsterdam, Netherlands: North-Holland.

Gould, J. D., and Boies, S. J. (1983). Human factors challenges in creating a principal support office system – the speech filing system approach. *ACM Transactions on Information Systems*, *1*, 273-298.

Gould, J. D., and Lewis, C. (1984). *Designing for usability: Key principles and what designers think* (Tech. Report RC-10317). Yorktown Heights, NY: International Business Machines Corp.

Gould, J. D., Boies, S. J., Levy, S., Richards, J. T., and Schoonard, J. (1987). The 1984 Olympic message system: A test of behavioral principles of system design. *Communications of the ACM, 30*, 758-769.

GraphPad. (2004). Confidence interval of a proportion or count. Available at http://graphpad.com/quickcalcs/ConfInterval1.cfm.

Gray, W. D., and Salzman, M. C. (1998). Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction, 13*, 203-261.

Greene, S. L., Gomez, L. M., and Devlin, S. J. (1986). A cognitive analysis of database query production. In Proceedings *of the 30th Annual Meeting of the Human Factors Society* (pp. 9-13). Santa Monica: CA: Human Factors Society.

Grimm, S. D., and Church, A. T. (1999). A cross-cultural study of response biases in personality measures. *Journal of Research in Personality, 33*, 415-441.

Hackman, G. S., and Biers, D. W. (1992). Team usability testing: Are two heads better than one? In *Proceedings of the Human Factors and Ergonomics Society 36th Annual Meeting* (pp. 1205-1209). Atlanta, GA: Human Factors and Ergonomics Society.

Harris, R. J. (1985). *A primer of multivariate statistics*. Orlando, FL: Academic Press.

Hassenzahl. M. (2000). Prioritizing usability problems: data driven and judgment driven severity estimates. *Behaviour & Information Technology, 19*, 29-42.

Hertzum, M., and Jacobsen, N. J. (2003). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction*, 15, 183-204.

Holleran, P. A. (1991). A methodological note on pitfalls in usability testing. *Behaviour & Information Technology, 10*, 345-357.

Ibrahim, A. M. (2001). Differential responding to positive and negative items: The case of a negative item in a questionnaire for course and faculty evaluation. *Psychological Reports, 88*, 497-500.

ISO. (1998). *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability* (ISO 9241-11:1998(E)). Geneva, Switzerland: Author.

Jeffries, R., and Desurvire, H. (1992). Usability testing vs. heuristic evaluation: Was there a contest? *SIGCHI Bulletin, 24*, 39-41.

Kantner, L. and Rosenbaum, S. (1997). Usability studies of WWW sites: Heuristic evaluation vs. laboratory testing." In *Proceedings of SIGDOC 1997* (pp. 153-160). Salt Lake City, UT: ACM.

Karat, C. (1997). Cost-justifying usability engineering in the software life cycle. In M. Helander, T. K. Landauer, and P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 767-778). Amsterdam, Netherlands: Elsevier.

Karat, J. (1997). User-centered software evaluation methodologies. In M. Helander, T. K. Landauer, and P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 689-704). Amsterdam, Netherlands: Elsevier.

Keenan, S.L., Hartson, H.R., Kafura, D.G., and Schulman, R.S. (1999). The Usability Problem Taxonomy: A framework for classification and analysis. *Empirical Software Engineering, 1*, 71-104.

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems, 2*, 26-41.

Kelley, J. F. (1985). CAL - A natural language program developed with the OZ Paradigm: Implications for supercomputing systems. In the *First International Conference on Supercomputing Systems* (pp. 238-248). New York: ACM.

Kennedy, P. J. (1982). Development and testing of the operator training package for a small computer system. In *Proceedings of the Human Factors Society 26th Annual Meeting* (pp. 715-717). Santa Monica, CA: Human Factors Society.

Kessner, M., Wood, J., Dillon, R. F., and West, R. L. (2001). On the reliability of usability testing. In Jacko, J. and Sears, A., Eds., *Conference on Human Factors in Computing Systems: CHI 2001 Extended Abstracts* (pp. 97-98). Seattle, WA: ACM Press.

Kirakowski, J. (1996). The Software Usability Measurement Inventory: Background and usage. In P. Jordan, B. Thomas, and B. Weerdmeester (Eds.), *Usability Evaluation in Industry* (pp. 169-178). London, UK: Taylor and Francis. (Also, see http://www.ucc.ie/hfrg/questionnaires/sumi/index.html.)

Kirakowski, J., and Corbett, M. (1993). SUMI: The Software Usability Measurement Inventory. *British Journal of Educational Technology, 24*, 210-212.

Kirakowski, J., and Dillon, A. (1988). *The computer user satisfaction inventory (CUSI): Manual and scoring key*. Cork, Ireland: Human Factors Research Group, University College of Cork.

Kraemer, H. C., and Thiemann, S. (1987). *How many subjects? Statistical power analysis in research*. Newbury Park, CA: Sage.

LaLomia, M. J., and Sidowski, J. B. (1990). Measurements of computer satisfaction, literacy, and aptitudes: A review. *International Journal of Human-Computer Interaction, 2*, 231-253.

Landauer, T. K. (1997). Behavioral research methods in human-computer interaction. In M. Helander, T. K. Landauer, and P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 203-227). Amsterdam, Netherlands: North Holland.

Lewis, C., and Norman, D. (1986). Designing for error. In D. A. Norman and S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction* (pp. 411-432). Hillsdale, NJ: Lawrence Erlbaum.

Lewis, J. R. (1982). Testing small system customer set-up. In *Proceedings of the Human Factors Society 26th Annual Meeting* (pp. 718-720). Santa Monica, CA: Human Factors Society.

Lewis, J. R. (1991a). A rank-based method for the usability comparison of competing products. In *Proceedings of the Human Factors Society 35th Annual Meeting* (pp. 1312-1316). San Francisco, CA: Human Factors Society.

Lewis, J. R. (1991b). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: The ASQ. *SIGCHI Bulletin, 23*, 78-81.

Lewis, J. R. (1992). Psychometric evaluation of the Post-Study System Usability Questionnaire: The PSSUQ. In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 1259-1263). Atlanta, GA: Human Factors Society.

Lewis, J. R. (1993). Multipoint scales: Mean and median differences and observed significance levels. *International Journal of Human-Computer Interaction, 5*, 382-392.

Lewis, J.R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors, 36*, 368-378.

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction, 7*, 57-78.

Lewis, J. R. (1996a). Binomial confidence intervals for small sample usability studies. In G. Salvendy and A. Ozok (eds.), *Advances in Applied Ergonomics: Proceedings of the 1st International Conference on Applied Ergonomics -- ICAE '96* (pp. 732-737). Istanbul, Turkey: USA Publishing.

Lewis, J. R. (1996b). Reaping the benefits of modern usability evaluation: The Simon story. In G. Salvendy and A. Ozok (eds.), *Advances in Applied Ergonomics: Proceedings of the 1st International Conference on Applied Ergonomics -- ICAE '96* (pp. 752-757). Istanbul, Turkey: USA Publishing.

Lewis, J. R. (2001a). Evaluation of procedures for adjusting problem-discovery rates estimated from small samples. *International Journal of Human-Computer Interaction, 13*, 445-479.

Lewis, J. R. (2001b). Introduction: Current issues in usability evaluation. *International Journal of Human-Computer Interaction, 13*, 343-349.

Lewis, J. R. (2002). Psychometric evaluation of the PSSUQ using data from five years of usability studies. *International Journal of Human-Computer Interaction, 14*, 463-488.

Lewis, J. R., Henry, S. C., and Mack, R. L. (1990). Integrated office software benchmarks: A case study. In D. Diaper et al. (Eds.), *Human-Computer Interaction - INTERACT '90, Proceedings of the Third IFIP Conference on Human-Computer Interaction* (pp. 337-343). Cambridge, England: Elsevier Science Publishers.

Lewis, J. R., and Pallo, S. (1991). *Evaluation of graphic symbols for phone and line* (Tech. Report 54.572). Boca Raton, FL: International Business Machines Corp.

Lilienfeld, S. O., Wood, J. M., and Garb, H. N. (2000). The scientific status of projective techniques. *Psychological Science in the Public Interest, 1*, 27-66.

Lord, F. M. (1953). On the statistical treatment of football numbers. *American Psychologist*, 8, 750-751.

Macleod, M., Bowden, R., Bevan, N., and Curson, I. (1997). The MUSiC performance measurement method. *Behaviour & Information Technology, 16*, 279-293.

Marshall, C., Brendan, M., and Prail, A. (1990). Usability of product X – lessons from a real product. *Behaviour & Information Technology, 9*, 243-253.

Massaro, D. W. (1975). *Experimental psychology and information processing*. Chicago, IL: Rand McNally.

Mayer, R. E. (1997). From novice to expert. In M. G. Helander, T. K. Landauer, and P. V. Prabhu (eds.), *Handbook of Human-Computer Interaction* (pp. 781-795). Amsterdam: Elsevier.

McFadden, E., Hager, D. R., Elie, C. J., and Blackwell, J. M. (2002). Remote usability evaluation: overview and case studies. *International Journal of Human Computer Interaction, 14*, 489-502.

McSweeney, R. (1992). *SUMI -- A psychometric approach to software evaluation*. Unpublished MA (Qual) thesis in Applied Psychology, University College Cork, Ireland.

Miller, L. A., Stanney, K. M., and Wooten, W. (1997). Development and evaluation of the Windows Computer Experience Questionnaire (WCEQ). *International Journal of Human-Computer Interaction, 9*, 201-212.

Moffat, B. (1990). Normalized performance ratio -- a measure of the degree to which a man-machine interface accomplishes its operational objective. *International Journal of Man-Machine Studies, 32*, 21-108.

Molich, R., Bevan, N., Curson, I., Butler, S., Kindlund, E., Miller, D., and Kirakowski, J. (1998). Comparative evaluation of usability tests. In *Usability Professionals Association Annual Conference Proceedings* (pp. 189-200). Washington, DC: Usability Professionals Association.

Molich, R., Ede, M. R., Kaasgaard, K., and Karyukin, B. (2004). Comparative usability evaluation. *Behaviour & Information Technology, 23*, 65-74.

Morse, E. L. (2000). The IUSR project and the Common Industry Reporting Format. In *Proceedings of the Conference on Universal Usability* (pp. 155-156). Arlington, VA: ACM.

Myers, J. L. (1979). *Fundamentals of experimental design*. Boston, MA: Allyn and Bacon.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *CHI '92 Conference Proceedings* (pp. 373-380). Monterey, CA: ACM.

Nielsen, J. (1993). *Usability engineering*. San Diego, CA: Academic Press.

Nielsen, J. (1994). Usability laboratories. *Behaviour and Information Technology, 13*, 3-8.

Nielsen, J. (1997). Usability testing. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics*. New York, NY: John Wiley.

Nielsen, J., and Landauer, T.K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of ACM INTERCHI'93 Conference* (pp. 206-213). Amsterdam, Netherlands: ACM Press.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces. *In Conference Proceedings on Human Factors in Computing Systems – CHI90* (pp. 249-256). New York, NY: ACM.

Nisbett, R. E., and Wilson, T. D. (1977). Telling more than we can know: verbal reports on mental processes. *Psychological Review, 84*, 231-259.

Norman, D. A. (1983). Design rules based on analyses of human error. *Communications of the ACM, 26*, 254-258.

Norman, D. (1986). Cognitive engineering. In D. A. Norman and S. W. Draper (Eds.), *User centered system design: New Perspectives on human-computer interaction* (pp. 31-61). Hillsdale, NJ: Erlbaum.

Nunnally, J.C. (1978). *Psychometric theory*. New York, NY: McGraw-Hill.

Palmquist, R. A., and Kim, K. S. (2000). Cognitive style and on-line database search experience as predictors of web search performance. *Journal of the American Society for Information Science, 51*, 558-566.

Parasuraman, A. (1986). Nonprobability sampling methods. In *Marketing Research* (pp. 498-516). Reading, MA: Addison-Wesley.

Prümper, J., Zapf, D., Brodbeck, F. C., and Frese, M. (1992). Some surprising differences between novice and expert errors in computerized office work. *Behaviour & Information Technology, 11*, 319-328.

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to Cognitive Engineering*. New York, NY: Elsevier.

Rengger, R. (1991). Indicators of usability based on performance. In H. J. Bullinger (Ed.), *Human Aspects in Computing, Design and Use of Interactive Systems and Work with Terminals, Proceedings of the Fourth International Conference on Human Computer Interaction* (pp. 656-660). Stuttgart, Germany: Elsevier Science Publishers.

Rosenbaum, S. (1989). Usability evaluations versus usability testing: When and why? *IEEE Transactions on Professional Communication, 32*, 210-216.

Rubin, J. (1994). *Handbook of usability testing: How to plan, design, and conduct effective tests*. New York, NY: John Wiley.

Ruthford, M. A., and Ramey, J. A. (2000). Design response to usability test findings: A case study based on artifacts and interviews. *IEEE International Professional Communication Conference* (pp. 315-323). Piscataway, NJ: IEEE.

Sadowski, W. J. (2001). Capabilities and limitations of Wizard of Oz evaluations of speech user interfaces. In *Proceedings of HCI International 2001: Usability Evaluation and Interface Design* (pp. 139-143). Mahwah, NJ: Lawrence Erlbaum.

Sauro, J. (2004). *Restoring confidence in usability results*. Available at http://www.measuringusability.com/conf_intervals.htm.

Shackel, B. (1990). Human factors and usability. In J. Preece and L. Keller (Eds.), *Human-Computer Interaction, Selected Readings* (pp. 27-41). Hemel Hempstead, UK: Prentice Hall International.

Shneiderman, B. (1987). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.

Smith, B., Caputi, P., Crittenden, N., Jayasuriya, R., and Rawstorne, P. (1999). A review of the construct of computer experience. *Computers in Human Behavior, 15*, 227-242.

Smith, D. C., Irby, C., Kimball, R., Verplank, B., and Harlem, E. (1982). Designing the Star user interface. *Byte, 7(4)*, 242-282.

SOGSC (Southwest Oncology Group Statistical Center). (2004). *Binomial confidence interval*. Available at http://www.swogstat.org/stat/public/binomial_conf.htm.

Steele, R. G. D., and Torrie, J. H. (1960). *Principles and procedures of statistics*. New York, NY: McGraw-Hill.

Stevens, S. S. (1951). Mathematics, measurement, and psychophysics. In S. S. Stevens (Ed.), *Handbook of Experimental Psychology* (pp. 1-49). New York: John Wiley.

Sudman, S. (1976). *Applied sampling*. New York: NY: Academic Press.

Turner, C. W., Lewis, J. R., and Nielsen, J. (2006). Determining usability test sample size. In W. Karwowski (Ed.), *International Encyclopedia of Ergonomics and Human Factors* (pp. 3084-3088). Boca Raton, FL: CRC Press.

van de Vijver, F. J. R., and Leung, K. (2001). Personality in cultural context: Methodological issues. *Journal of Personality*, *69*, 1007-1031.

Virzi, R. A. (1990). Streamlining the design process: Running fewer subjects. In *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 291-294). Santa Monica, CA: Human Factors Society.

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

Virzi, R. A. (1997). Usability inspection methods. In M. G. Helander, T. K. Landauer, and P. V. Prabhu (eds.), *Handbook of Human-Computer Interaction* (pp. 705-715). Amsterdam: Elsevier.

Virzi, R. A., Sorce, J. F., and Herbert, L. B. (1993). A comparison of three usability evaluation methods: Heuristic, think-aloud, and performance testing. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting* (pp. 309-313). Santa Monica, CA: Human Factors and Ergonomics Society.

Vredenburg, K., Mao, J. Y., Smith, P. W., and Carey, T. (2002). A survey of user centered design practice. In *Proceedings of CHI 2002* (pp. 471-478). Minneapolis, MN: ACM.

Walpole, R. E. (1976). *Elementary statistical concepts*. New York, NY: Macmillan.

Wiethoff, M., Arnold, A., and Houwing, E. (1992). *Measures of cognitive workload* (MUSiC ESPRIT Project 5429 document code TUD/M3/TD/2).

Wenger, M. J., and Spyridakis, J. H. (1989). The relevance of reliability and validity to usability testing. *IEEE Transactions on Professional Communication*, *32*, 265-271.

Whiteside, J., Bennett, J., and Holtzblatt, K. (1988). Usability engineering: Our experience and evolution. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 791-817). Amsterdam: North-Holland.

Wickens, C. D. (1998). Commonsense statistics. *Ergonomics in Design*, *6(4)*, 18-22.

Wildman, D. (1995). Getting the most from paired-user testing. *interactions*, *2(3)*, 21-27.

Williams, G. (1983). The Lisa computer system. *Byte*, *8(2)*, 33-50.

Wixon, D. (2003). Evaluating usability methods: Why the current literature fails the practitioner. *interactions*, *10(4)*, 28-34.

Woolrych, A., and Cockton, G. (2001). Why and when five test users aren't enough. In Vanderdonckt, J., Blandford, A. and Derycke A., (Eds.), *Proceedings of IHM-HCI 2001 Conference, Vol. 2* (pp. 105-108). Toulouse, France: Cépadèus Éditions.

Wright, R. B., and Converse, S. A. (1992). Method bias and concurrent verbal protocol in software usability testing. In *Proceedings of the Human Factors and Ergonomics Society 36th Annual Meeting* (pp. 1220-1224). Atlanta, GA: Human Factors and Ergonomics Society.

Wright, P. C., and Monk, A. F. (1991). A cost-effective evaluation method for use by designers. *International Journal of Man-Machine Studies, 35*, 891-912.