Mis-Usability: On the Uses and Misuses of Usability Testing

R. Stanley Dicks
North Carolina State University
Department of English
Raleigh, NC 27695-8105
(919) 515-4124
sdicks@unity.ncsu.edu

ABSTRACT
Usability has become so popular that its value is being threatened by misuse of the term and by misunderstandings about important distinctions between usability studies and empirical usability testing, between usability and verification tests, between ease of use and usefulness, and concerning how we interpret statistics. This paper examines these problems and encourages academicians and practitioners to work toward mitigating them.

Categories and Subject Descriptors
H.1.2 **User/Machine Systems**: Human Factors

General Terms
Documentation, Performance, Design, Human Factors.

Keywords
Usability, Documentation

1. INTRODUCTION
Usability testing may be on the verge of becoming a victim of its own success. Those of us who have expounded on the virtues and necessities of usability testing are now watching as people apply the concept, the methods, and the results in ways we could not have imagined. Automated testing programs are "testing" websites and yielding reams of tables, charts, and graphs of "usability" information. People are usability testing anything that moves, and, in some cases, things that neither move nor have any usable characteristics. The web is filled with sites offering lists of web design "rules," sometimes based on testing performed with a handful of subjects. One of the seminal web testing books is based on testing using flawed methods on a non-representational set of users doing non-representational tasks. This is a problem for technical communicators, who are increasingly being expected to perform usability testing on their documents, help systems, and websites. Indeed, as more and more of technical communicators' products are delivered online, and hence are software, we will undoubtedly be asked to test those products. Many communicators are being asked to learn usability testing methods quickly, and, in some cases, inadequately.

It has been difficult enough to persuade most managers to fund usability testing of products under development. If we allow tests that do not take into account the inherent limitations of usability testing and if we allow poor test methodologies to yield questionable results (as indeed they will), we risk undermining and trivializing the whole concept of usability testing. That risk becomes more severe if we allow others to appropriate usability testing methods without really understanding what they are doing.

Four general categories of misconceptions contribute to mis-usability:

1. Misunderstanding of the concept of usability itself and of the distinctions between usability studies and empirical tests.
2. Two types of problems with statistics: assuming that a set of quantitative statistics equals a usability test, and misusing statistical results, especially from tests performed without large enough user samples.
3. Using usability tests for verification rather than usability.
4. Lack of knowledge of the limitations of and the proper methods of usability testing to ensure valid and reliable results.
5. Testing for ease of use but not usefulness.

This presentation examines each of these misconceptions. It will provide examples of the conceptual problems and will list the major limitations of usability testing that all practitioners should take into account.

SIGDOC'02, October 20-23, 2002, Toronto, Ontario, Canada.

2. THE CONCEPT OF USABILITY
What is usability? Perhaps more importantly, what is not? Dumas and Redish [2] define usability testing as requiring five characteristics: that the goal is to improve a product's usability, that the participants represent real users, that they do real tasks, that testers observe and record the participants, and that they then analyze the data and recommend changes to fix problems.

Most usability experts include four aspects of usability, taken from Gould and Lewis [3], in their conception of the term:

1. easy to learn
2. useful
3. easy to use
4. pleasant to use

Two main problems arise associated with the overall concept of usability. First is an attribution or semantic problem. Some people claim that they are performing usability tests on artifacts that are not usable. After the Florida ballot controversy in 2000, usability testing began to develop some cachet. At the same time, more and more programming books and articles were stressing the importance of doing usability tests on both systems and documentation, and of doing so early in the design process. Consequently, people have begun using usability testing techniques on a variety of artifacts.

In one case, for example, I read of a college course being usability tested. Is a college course usable? It may have components, such as a website, a syllabus, or a textbook, that can be tested, but a course itself cannot. Usability testing methods have been developed and refined over many years to allow testing of artifacts that have usable characteristics, what Norman [8] calls affordances and constraints. Employing those same test methods on non-usable subjects may give very misleading results. And using a few usability methods, such as the "talk aloud" method, does not mean that one is performing a usability test. Having students "talk aloud" about what they liked and didn't like in a college course is not a usability test. It is more on the order of a market assessment.

The second type of conceptual usability problem involves the distinction between gathering usability data and performing empirical, reproducible usability tests. Rubin [9] and Barnum [1] distinguish between formal tests conducted using the experimental method to prove or refute hypotheses and less formal tests used to discover and correct usability problems with a product. The more formal method requires large sample sizes of participants, careful test construction and implementation, and analysis of inferential statistics to arrive at the validity and reliability of the results. Usability adherents quickly discovered that most product development managers are simply not willing to pay the price in time and money that such testing requires. As a result, Nielsen [7] advocated "discount" usability testing, showing that tests with as few as four or five users are able to uncover 80% of the problems with a product. Such tests do fit within most development budgets and also allow for earlier, more valuable, and more frequent testing to be done. However, they also lead to some confusion about discovering usability problems and conducting formal, empirical, repeatable research.

The results of a typical usability test are "good enough" to help us uncover problems with a product and to correct enough of those problems so that the testing more than pays for itself. While tests with small sample sizes do not discover all of the problems, iterative testing can usually catch most of them. It is important to note, though, that such tests do not provide us with valid or reliable results for any kind of data, including usability, user preferences, marketing data, etc. Because the tests are good enough for usability purposes, people not versed in the scientific method are prone to assuming that the results of such tests can be given the same credit as those from more formal testing. They cannot.

3. LIES, DAMNED LIES, AND STATISTICS

The exponential growth of the Web and the number of commercial sites on it has led to the development of dozens of programs designed to monitor and report statistics related to users' interaction with the site, including such measures as the number of site visits, the screens visited, tasks completed, tasks not completed, site abandonments, etc. These monitoring capabilities are often referred to as automated "usability tests."

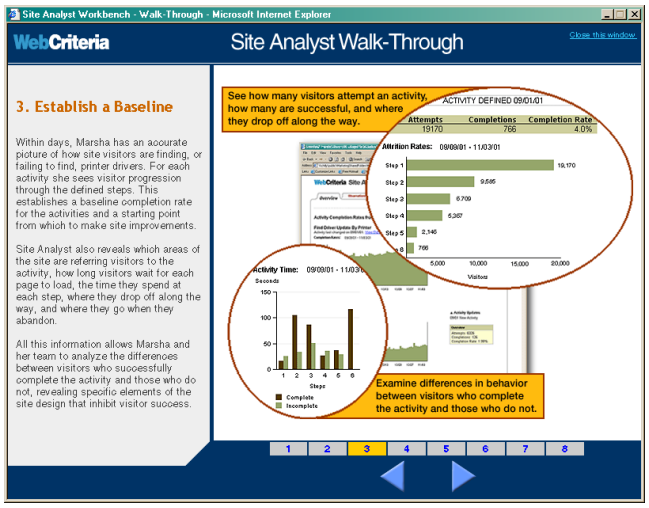

Figure 1. WebCriteria Report Example

The WebCriteria suite of measurement tools provides an example. As Figure 1 shows, reports can offer a variety of statistics and graphics aimed at helping organizations learn how users are interacting with their sites. Such information can be quite valuable. However, it is not, in the strictest sense, usability data. If we consider the four functions that usability tests measure—ease of learning, usefulness, ease of use, pleasant to use—such tools do not tell us which of these usability characteristics are behind a reported problem. If, for example, our monitoring software tells us that half our users abandon the site when they get to page 12, we know there is a problem on page 12, but we have no idea which of the usability characteristics page 12 is not doing well. Consequently, we know we need to do something on page 12, but we do not know what. Conversely, with the combined quantitative and qualitative data derived from a usability test, we know where the problem is and, usually, we have a reasonably good idea about how to fix it.

Comprehensive printouts of colorful graphs, charts, and reports may provide us with some helpful data about where usability problems are occurring, but they do not substitute for the more valuable information gained from actually conducting tests. Further, such testing software is generally designed to work with completed applications or sites, so it may not provide us with the more valuable results we get from doing usability testing earlier in the design process.

A second problem associated with statistics comes about because many people, even highly educated ones, do not understand the distinction between the quantitative and qualitative data collected in most usability tests and the purely quantitative data reported in more rigorous, empirical, research studies. They either make the mistake of thinking that the data in a usability test must be completely worthless because the sample size is too small, or they make the converse assumption that the results of the test offer universal truths.

After teaching usability courses for several years, I have been alarmed at the tendency among graduate students, no matter how often I distinguish between formal and informal usability tests, to make sweeping statements about how their test results have "proven" that a certain product feature is loved or loathed by all users. As Hughes [4] points out, unsophisticated practitioners and clients are likely to interpret the mixture of quantitative and qualitative data yielded by a discount usability test by using the assumptions attached only to quantitative data. For example, if three out of four users on a test have trouble with feature z, we use that quantitative result to glean where to look at the qualitative data, such as the user's reports while trying feature z and our own video, audio, and notes about what happened as users tried feature z. Doing so guides us to places in the product where there are most likely to be usability problems. However, we cannot say with any confidence that three out of four users in the entire user population will have trouble with feature z.

Dumas and Redish [2], Rubin [9], and Barnum [1] all warn about the problems that most people have understanding inferential statistics. Nonetheless, many people who should know better continue to make pronouncements about universal web usability based on inadequate sample sizes. In fact, there seems to be an inexplicable desire of people to make such pronouncements even when they should know better. Spool et al [10] published one of the first books about web usability, but they did not test with large enough sample sizes to warrant the general claims they make throughout the book. Even Nielsen, on his website (http://www.useit.com/) and in his book *Designing Web Usability* [6] often makes what frequently seem to be general claims based on testing that has been done with small samples. If anyone is entitled to proffer opinions about usability and design standards, it is Nielsen. But it is important to note that they are often opinions and not based on the results of tests performed with statistically valid and reliable sample sizes.

4. Usability vs. Verification Testing

Many product developers believe, through years of training and practice, that testing is something you do only upon completion (or near completion) of the development cycle. The verification testing done with software must be performed late enough in the cycle so that any changes that come as a result will not affect the software enough to require another complete round of verification testing. The same is true with quality and verification tests done on hardware.

Likewise, it is fairly standard practice to perform tests against documents and online systems at the conclusion of a development cycle. These tests are sometimes called final edits, quality checks, or verification tests (especially if they are done in conjunction with the product). Some documentation groups have begun to add a few usability criteria to the list of things they check and to refer to these final tests as usability tests. This gives everyone involved a cheap, easy was to say that they did usability testing, but it is a subversion of the entire idea of usability. While human factors experts can perform heuristic evaluations that uncover some usability problems, the kind of checklist testing done by managers, editors, or testers at the conclusion of a development process should not be called usability testing. To do so cheapens the whole concept of usability, which requires that we test with "real users."

Some programmers and engineers mistakenly believe that if you test a document against a piece of software, and that if you confirm that all of the procedures in the document actually work, then you have demonstrated the usefulness (and hence the usability) of the document. They are happy, then, to call such a test a usability test. When this happens, technical communicators must educate their development peers about what usability actually means.

5. Inherent Limitations of Usability Testing

Usability testing, whether it is formal or informal, has a number of inherent shortcomings. These have been frequently discussed, but it is worth mentioning them again, as many people who have learned the procedures for usability tests sometimes overlook the limitations. As Rubin [9] puts it, "Even the most rigorously conducted formal test cannot, with 100 percent certainty, ensure that a product will be usable when released" (p. 27). He cites four main reasons why:

1. Testing is always an artificial situation.
2. Test results do not prove that a product works.
3. Participants are rarely fully representative of the target population.
4. Testing is not always the best technique to use.

As discussed previously, nearly all usability testing is done with smaller than statistically significant sample sizes, and it is often done with less than rigorous methods. Further, even fairly extensive testing may not uncover major problems with a product. We have all struggled to decide which tasks to test with a large, complex piece of software. For most tests, we can only choose a small, selected group of tasks to test, given the limitations of time and resources for testing. What we try to do is to choose tasks that will represent the product as a whole, with the assumption that if users can successfully complete our selected tasks that they can complete all other tasks using the product. Most of us who have done enough usability testing have experienced cases where that assumption did not prove true.

Indeed, the task orientation of usability testing also ensures some of its limitations. Focusing so heavily on tasks can cause us to design tests that do not uncover larger, more global problems, especially related to how users conceive of the overall product and its processes, what their mental maps are. We can design tests that show that users successfully completed many small tasks, but we may have masked the fact that they cannot understand how to construct larger, more complex tasks or to understand the overall operation of the software. This is particularly true related to documentation and to online systems. It is easy to design a test that shows that users can complete tasks using a manual or a help

system. It is much more difficult to design and implement a test that shows users stringing together complicated sets of smaller tasks to achieve larger goals.

There is a further problem with the qualitative parts of a usability test, especially related to what I like to call preference data. I have also seen this called 'attitude' or 'likability,' although I cannot bring myself to use the latter. Participants in usability tests are often in a strange environment. They may make many assumptions about what is going on that are not accurate, including the possibility that they feel compelled to impress you and to under report errors. We have all seen cases where participants struggled through several parts of a test and then reported at the end that the product was easy to use and that they had no major problems. Likewise, we have seen users breeze through every part of a product, but then report that they still did not like it. While we often test how well users can perform a small group of tasks, we may not be testing effectively enough how well they like the overall product design (especially related to web tests) or how well they appreciate the product's ethos, their sense of its authority and quality.

The task orientation so often promulgated in the technical communication literature contributes to this problem. It is far easier and seems to be more in line with what the experts say to test a group of representative tasks than it is to try to test more difficult product aspects, such as usefulness, attitude, or ethos. It is easy to fool ourselves into believing that testing a few ease of use tasks equals assuring the overall usability of a document, an online system, or a product. This brings us to the final and most significant problem with most usability testing.

6. Ease of Use versus Usefulness

Two of the four items on the original Gould and Lewis [3] list of usability characteristics are 'ease of use' and 'usefulness.' To many people, these sound redundant. However, there is a critical distinction to be made. As these terms have most often been interpreted, ease of use refers to efficiency, to how quickly we can use a product to complete tasks. Usefulness refers to the overall usefulness of the product. Does it do what it is supposed to? Is it usable at all? Does it work? The distinction is important because it is possible to make a product that is less than the sum of its parts. While each part (or task) may work fine by itself, as we add more and more of them, we get a geometrical increase in the number of relationships and possible interactions among them. We also get an increase in the number of possible methods for completing a given set of tasks to perform some larger, more complex task. Further, such systems are often used in environments that are changing constantly, where new processes and procedures are being introduced to the environment constantly. The overall system, then, needs to have the flexibility to allow users to complete tasks in ways that might not have been considered during original design. The overall system needs to be designed with usefulness at the first usability criteria rather than as one that is assumed to be met when a series of ease of use criteria are met.

Most usability tests establish several tasks for users to complete, including reasonable times required for their completion. We may be certain that we have selected a set of typical user tasks, based, perhaps, on field observations of users doing their jobs. We then construct a test that measures how well users complete the tasks within designated time periods. The result is a reassuring set of data that shows us that we have a usable product. However, as Mirel [5] has pointed out, that assurance may be very misleading. With large, complex systems, we may not know whether users understand the overall complexity of the system, the relationships of its various parts, methods for constructing complicated tasks using those various parts, or whether some of the functions we did not test are deficient, leading the overall system to failure even when many of its parts work individually. Mirel uses the example of a medication software system that works fine for the small task of administering drugs to a patient, but that fails to connect to other databases and hospital functions that the users (nurses) very much need to make the system truly useful. A typical usability test on the medication software itself, measuring success in administering to the needs of a single patient, would demonstrate that the software has sufficient ease of use. However, a larger test designed to test its overall usefulness in the complete context in which it is used, would show significant problems.

I have seen this problem in website testing that we have conducted in the last two years. With some types of sites, our results have shown that users can complete every task assigned on a test within reasonable times. However, they often report that they do not like the overall site and would not use it again. Some of that may be due to esthetics and/or to brand preferences, but it has often been due to larger, overall usefulness problems associated with the sites. While it might have been "easy" to complete smaller tasks, using the site to do anything complicated or out of the ordinary was simply too difficult.

Mirel issues a call for the usability community to reform itself and to re-focus on usefulness at the primary usability criterion:

> Usability leaders are needed for articulating the primacy of usefulness and for shifting the unit of analysis for design and development from task actions to task structure—to the structural arrangements and relations between people, resources, and contextual conditions for a given task or problem. (p. 182)

7. Conclusion

Usability becomes an increasingly more important concept as technology allows us to connect more and more things, people, and sets of data. We must not allow it to be redefined and co-opted so that verification tests are thought to measure usability. We must constantly educate those in other areas as to the meaning of usability and to its various characteristics as described by Gould and Lewis and others, not simply that it refers to ease of use. We must challenge those who make universal claims based on inadequate evidence for doing so. And we must look at overall usefulness of products instead of focusing so intently on their interfaces and their smaller elements.

As more and more university-level, technical communication courses and texts have usability components added to them, it becomes very important that professors discuss the limitations of usability testing and that they introduce at least enough information about inferential statistics so that people do not believe that they are demonstrating universal truths based on sample sizes of four users. It is also important that they discuss all of the usability characteristics, rather than focusing too heavily on ease of use, task oriented testing for procedural information.

Practitioners need to challenge those who attempt to reduce usability to something less than it actually is, whether that means mistaking it for verification or applying it too thinly to small sets of product features. We also need to insist that those who offer design guidelines do so based on adequate tests using real users doing real tasks.

None of this should thwart practitioners from continuing to gather usability information. Even though the results of using one of the "discount" usability methods or of using small samples may not stand up to the rigors of controlled experimental enquiry, they can still yield very useful benefits to practitioners developing computer documentation. As Rubin [9] points out, "*…it is better to test than not to test*" (p. 27). Limited testing may not verify with absolute certainty that a product is useful and may not prove larger hypotheses, but it can yield results that allow us to make major improvements to products before they are delivered to customers.

Most documentation development groups must fight for both the time and the resources to conduct any usability testing at all. Given the realities of fiscal and schedule restraints, very few of us get to perform full-scale usability testing with large sample sizes from our audience. So, we must rely on "discount" methods and on small sample sizes. Doing so can still help us find and correct major and minor problems with our products and to release greatly improved paper and online documentation. As long as we do this without falling into the errors of claiming that we have *verified* usability or *proven* larger concepts, we will not have fallen into the trap of mis-usability.

8. REFERENCES

[1] Barnum, C. Usability Testing and Research. Longman, New York, 2002.

[2] Dumas, J. S., and Redish, J. C. A Practical Guide to Usability Testing. Ablex, Norwood, NJ, 1994.

[3] Gould, J. D. and Lewis, C. Designing for usability: Key principles and what designers think. Communications of the ACM, 28, 3 (March 1985), 300-311.

[4] Hughes, M. Rigor in usability testing. Technical Communication, 46, 4 (November 1999), 488-494.

[5] Mirel, B. Advancing a vision of usability in Reshaping Technical Communication (Mirel, B. and Spilka, R., eds.). Lawrence Erlbaum, Mahwah, NJ, 2002, 165-187.

[6] Nielsen, J. Designing Web Usability. New Rivers, Indianapolis, IN, 2000.

[7] Nielsen, J. Guerilla HCI: Using discount usability engineering to penetrate the intimidation barrier in Cost Justifying Usability. (Bias, R. G. and Mayhew, D. J. eds.). Academic Press, Boston, 1994, 242-272.

[8] Norman, D. The Design of Everyday Things. Doubleday, New York, 1988.

[9] Rubin, J. Handbook of Usability Testing. Wiley, New York, 1994.

[10] Spool et al. Web Site Usability: A Designer's Guide. User Interface Engineering, North Andover, MA, 1997.