Deborah J. Mayhew & Associates

Usability Testing - You Get What You Pay For

By Deborah J. Mayhew, PhD

I have been conducting usability tests of software and website user interfaces for many years, using a methodology based on what I learned in a graduate program in experimental cognitive psychology, adapted to an engineering environment.

Over the years, I have encountered a lot of misconceptions among my client organizations regarding what constitutes a valid and useful test. For example, they often equate them with demos or focus groups, and/or believe that pretty much any one can plan and run one.

As I have talked and worked with usability professionals and others conducting usability tests, I have found that untrained and inexperienced testers make a variety of errors in the way they conduct usability tests, which probably account for the lack of reliability described in a Forrester report (Souza, 2002).

A valid and useful usability test takes expertise. A lack of rigorous controls invariably reduces the validity and usefulness of the data generated by the test. Here I offer a description of some of the key ingredients of a good usability test (for more detail, see Mayhew, 1999), and some examples of poorly designed or run testing.

SAMPLING

Sampling refers to selecting participants to be test users in the test. The main issues here are how many test users to include in a test, and what type of users to include. Most trained and experienced usability professionals will tell you that it is not necessary to run more than 6 to 8 test users in a particular test. While such a sample size will usually not yield results of statistical significance, most of us have found through experience that regardless of how many users you test in a given iteration, you see most of the glaring flaws in your design in the first 6-8 users you run. Statistical significance is required to establish scientific truths. It is not necessary to conduct good engineering.

However, what is critical is that your 6-8 test users be truly representative of the target user population (i.e., have the same set of key characteristics, such as job experience level, computer literacy level, age, gender, expected frequency of use, etc.) If you don't

88 Panhandle Road
Post Office Box 248
West Tisbury, MA 02575
PH: 508-693-7149
FX: 508-693-9726
Email: drdeb@vineyard.net
Website: http://djmassoc.com

know what the key characteristics of your target user population are, you cannot sample representatively. Thus, you must have in hand an accurate user profile in order to design a valid test.

Also, if you have multiple distinct user categories, then you must test 6-8 of each type. For example, if you expect doctors, nurses, technicians and receptionists to all use the same medical office software, you cannot test with just doctors and then assume that the other three user categories will respond in the same way to a given design. You also cannot run a total of 6-8 users, with only 1-2 users of each type - that is not a large enough sample of each type.

So, small numbers of users in each significant user category should be enough to reveal basic, showstopper flaws in a design. However, a small sample will be sufficient only if the rest of the test is well designed. A poorly designed and run test may indeed take 80 users to uncover 85% of flaws, and inexperienced testers may indeed draw very different conclusions watching the same test (Souza, 2002). Thus, a small sample is completely acceptable, but only if it is representative, and if all other aspects of the test - described below - are adequately addressed.

Case in point - An untrained tester in one vendor organization I worked in ran a test in which he recruited developers in his organization to be test users. The intended audience, however, were non-technical business people. Since he tested non-representative users, he really did not learn anything useful from his test.

Case in point - I ran a test of a web site once with a total sample of 8 test users. The first user flew through the test tasks with few errors. The next 7 users all made significant errors, experienced significant confusion and took much more time. Clearly any conclusions drawn from the first 1 or 2 users would have been misleading or inconclusive. On the other hand, 8 users produced an extremely clear pattern. More users would not have been necessary to validate this pattern.

TEST TASK DESIGN

Conducting a valid test also requires conducting a mini requirements analysis, in order to understand the real tasks that real users will be attempting to accomplish on the web site or software application being tested. If you don't thoroughly understand the actual tasks that users are likely to want to perform, you will not be able to design test tasks that represent those tasks. If you don't test the right tasks, you will not find important usability flaws.

For example, on a shopping web site, you might assume users will want to browse for products before buying, and so you might construct test tasks that involve first browsing for and selecting products, then completing a purchase. However, if users have usually just referred to a catalog and know exactly what product they want, then you will have tested the wrong task, and won't have generated useful data about the usability of your

site. In addition, you can rarely test all user tasks in a single test iteration, and if you don't know which are the key tasks and instead you test only less frequent or less important tasks, you will likely fail to find the more important usability flaws in a design.

Another reason to fail to find the main usability flaws, even with large numbers of test users, might be that test tasks are not structured enough. For example, if you simply ask users to explore a web site, they may or may not navigate to certain parts of the site where problems exist. You must design tasks in a way that insures that users will navigate (or at least try to navigate) to parts of the site you want data on. And, all users in your sample must navigate to those parts, so you have enough data points on interaction with those parts to draw valid conclusions. Again, you need to know what the important tasks are, and what pages or interactions are key to those tasks, and then insure that users attempt to perform those tasks, navigate to those pages and use those interactions, in order to test the right parts of the interface and thus reveal any important flaws.

Besides testing the right tasks, tasks must be presented to users in a way that does not lead them, mislead them or "give away" correct interactions. For example, if you have used the concepts and terms "shopping basket" and "checkout" on your shopping web site user interface, and in your test tasks you ask users to "get product X into your shopping basket" and "check out", you have mostly just tested the user's ability to read, rather than the effectiveness and intuitiveness of your user interface design. You need to express the task in clear terms that the user will understand, but without making direct reference to any aspect of the user interface, including labels and terminology. For example, in this case, you might use the phrases "add product X to your list of things to buy", and "complete your purchase." Then you will find out if the terminology and concepts in your interface are clear to users.

Testing the right tasks and being scrupulous in not introducing bias in the way you present tasks will make all the difference in whether you uncover actual usability flaws in your design. Understanding real and key tasks, and designing the presentation of those tasks to avoid introducing bias, requires experience and expertise.

Case in point - An untrained tester in one vendor organization I worked in ran a test of an input device to try and determine whether it would be effective as the main input device for a particular application under development. However, while the application being developed was a word processor, he had user's conduct simple graphical tasks (e.g., drawing, hitting graphical targets) with the device. As a result, he learned nothing useful about the device's efficacy for the application of interest, as input devices are often usable for some types of tasks but not others.

Case in point - I have over and over again observed or heard of testers who do not take the trouble to phrase tasks in ways that do not refer directly to UI terminology, concepts and organizational structure. In these tests, important flaws are undoubtedly missed because the way test tasks are presented in effect teaches the user how to use the user interface.

CONDUCTING THE TEST

A great deal rides on exactly how the test is conducted. Besides phrasing the test tasks in the right way (see above), in a moderated test, everything about how the tester interacts with the test user while they attempt to accomplish the task will affect the data generated. It is extremely important for the tester not to lead the user in any way, either through words or body language. Inexperienced testers unconsciously help users figure things out. The instinct to teach and help is very strong. It takes a lot of experience to become a truly neutral tester. When a user is really stuck, there are ways to give minimal hints to get them going again and still collect useful data. This too takes training and experience. When a test fails to reveal significant usability flaws, it is often due to inexperienced testers inadvertently leading users to correct interactions after presenting the task.

Case in point - In one moderated usability test I had a small group of observers from the development team present in the same room with the test users. I had to ask them over and over again to refrain from interrupting to talk with my test users. They could not contain their desire to explain the application to them, discuss design issues and explain why design decisions were made, and simply did not seem to understand how their interaction with the user was biasing the data being collected.

Case in point - In another usability test, halfway through the test users, I switched places with my assistant, having him moderate the test while I collected data. I was dismayed to observe that even after watching me moderate for half the test, my assistant inadvertently led the user to correct interactions in many cases.

CAPTURING THE DATA

In moderated tests, I usually rely on very detailed data collection sheets that allow me to capture the maximum amount of detailed data with the least amount of note taking. A number of times I have had assistants also collect data on my data collection sheets, and when I compare my data with theirs, I have usually noticed that not enough data - and different data - has been captured by my less experienced assistants. If data is not correctly and sufficiently recorded - even if everything else is done right - you will not end up with valid conclusions.

Case in point - In one case I had members of my client organization run a test I had designed. After the test they sent me both their data collection sheets and videotapes of the test sessions. I found that their data collection sheets failed to record a great deal of significant data in the test sessions that revealed important problems in the UI design.

INTERPRETING THE DATA

I have also noted when working with less experienced assistants that we interpret the same data quite differently. Interpreting test data correctly and effectively requires an understanding of human perception and cognition and a lot of design and testing experience. Like everything else about usability testing, it should not be left in the hands of novices.

Case in point - In one test, a problem revealed by the data was that users usually clicked on an incorrect button on a particular page. My assistant's interpretation was that the button needed a new and clearer label. Mine was that the whole page looked too similar to another page representing different functionality, and that the users were confused about which page they were on and what they were doing. A change in button labeling would not really have solved the underlying problem, which was perceptual in nature.

Case in point - In another example, an assistant wanted to solve a common problem test users had—finding things on a web site—by simply adding a general Search field. My interpretation was that there was a fundamental flaw in the information architecture.

Given the many places an untrained or inexperienced tester can go wrong, and the fact that going wrong in any one of them can invalidate results, it is not surprising that even tests including many test users do not uncover all design flaws, or that different testers find different problems, or that general testing does not really seem to improve user interface designs (Souza, 2002). It takes a well-designed and run test to produce reliable results, and an experienced data interpreter to apply those results effectively.

Organizations shopping for testing vendors should carefully assess the qualifications of the testers offered by those vendors. Also, be wary of anyone insisting that more than an average of 10 test users in a given category for a given testing iteration are necessary. This simply should not be necessary when the test is designed and run by a competent professional.

Usability testing is an invaluable technique, and it's worth spending a significant amount of money on. But there is little point in spending even a little money on a test that will not yield valid results. Development organizations shopping for usability testing vendors should look for evidence that a vendor has the required skills and experience to plan and run a valid usability test, and effectively interpret and utilize the results. Like most things, when it comes to usability testing, you get what you pay for.

REFERENCES

Souza, Randy, "Best Practices for Usability Testing", Forrester Report, May 2002

Mayhew, Deborah J., The Usability Engineering Lifecycle, Morgan Kaufmann Publishers, 1999 (a detailed description of the kind of rigorous UE program referred to in the article above)