The Software Quality Advisor Online

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX
February 2001 Newsletter

Web Usability Validation

by Randy Rice, CQA, CSTE

While most people agree that usability testing is important, usability experts also tell us that usability testing should be a validation that good usability factors have been designed into an application, not the initial point of identifying usability problems. Web applications are no different in this regard.

In this article I will cover the need and importance of usability testing for web applications, methods and processes for web usability testing, and how to measure and tweak your processes once you have them in place.

Web Usability Testing and Why We Need It

First, let's talk about usability testing in general. Usability testing is the validation that someone in the expected user audience can use something without extensive documentation or assistance. In the world of software, usability testing is a behavior test that measures ease of use factors, which can often become very subjective. Usability is also a kind of testing that takes extended observation to evaluate. This means that you can't just sit down and perform a few tests to determine usability, although you can get a pretty good idea of initial usability when you first start to perform basic functions. You can't automate usability testing since the human-computer interaction is the point of the entire exercise, therefore it can be a labor-intensive exercise.

Now, enter the web

In using PC-based shrink-wrapped software, such as Microsoft Office, the customer/user installs the software, hardly ever reads the manuals, and starts right to work doing something. If they have a problem, they click on the Help menu and if they are successful, they find the answer to their problem. In this case, the user is motivated to get the issue resolved because they have spent money on the product. To abandon the product and try something else would be a waste of time and money. They can't just take the software back because they find it confusing or difficult to use – there is no such warranty on most software, except in the cases of trial offers.

Software companies have for years employed usability test labs for the purpose of validating ease of use factors in their products. Some companies have relied on beta testers, but the ones who are serious about usability have a process and people to perform this kind of testing.

Web applications bring a new dimension and concern to software usability in the following ways:

- *The applications can change much more frequently than shrink -wrapped software.*

Web applications are often portals that affect a company's income stream, not by the sale of the application, but sales of other products. Having a confusing e-commerce web site is like walking into a store where you can't find what you need.

- *The audience for a web application can be diverse and hard to predict.*

New web development techniques and technologies are constantly evolving, which may or may not be user-friendly. An example of this is the "mouse-over" functions which allow the user to see details about something only when the mouse passes over it. This is also called "mystery meat," in that you don't really know what's under the gravy until you dig into it.

The web is a non-standardized and inconsistent place. When you surf the web you have to adjust to each web site's way of doing things. You wind up asking "Why couldn't these people do things like (fill in with your favorite easy to use web site)?"

- *People will not hang around a web site that is hard to use.* They can simply click over to a competitor's site.

So, if your motivation for being on the web is to serve your customers well and to make money, web usability should be a critical success factor.

What Do The Surveys Show?

Findings from the 1998 Graphics, Visualization and Usability (GVU) 10th WWW User Survey[1] indicate that:

- "Contrary to popular belief about the importance of security to consumers using the web for shopping, our survey showed that, overall, quality information, easy ordering and reliability were important to more respondents than security."

- "Overall, respondents reported dissatisfying experiences because of a number of factors. The main ones were sites were confusing/disorganized (74.5% of individuals and 22.7% of responses out of a possible 7); failure to find what they were looking for (71.3% of individuals and 21.7% of responses); and slow download (59.9% of individuals and 18.3% of responses). However, a surprising 57.1% of novices said that this had not happened to them yet (36.4% of responses)."

- "On average, respondents were spending 10-30 minutes searching before giving up if they were unable to find what they were looking for."

I conclude from these findings and the findings from our testing at Rice Consulting Services that:

- Ease of use factors are an essential ingredient to meeting your customers' needs on your web site.

- Some people will dig through a mess to find a bargain (just like in a real store!).

- People will eventually give up and move on if they can't find what they are looking for.

Processes, Tools and Techniques for Web Usability Testing

The process for performing web usability testing doesn't have to be complex. In fact, just like the web pages we are testing, the simpler the better.

Step 1 – Define the Audience

This is the most important thing you must get right for web usability testing. If you are designing a web site to be used by people over 60, your usability testers should be in that demographic. If you don't know people in that demographic, then do whatever it takes to find them or hire them. People have found that it doesn't take a lot to entice people to be usability testers. There is something within people that makes them excited and motivated to have their opinion queried – as long as the query is not intrusive or overly time-consuming. People need to be compensated in some way for their efforts. One example is the case where a dot com company was looking for usability testers in the 13 to 18 year-old demographic. They found kids who would "buy" a variety of products from their web site and then compensated them by giving them one of the items they ordered (up to a certain limit). This realized three things for the company: 1) realistic testers, 2) a good idea of what kids that age like, and 3) a good idea of how a typical customer will navigate through the web site.

In addition to profiling the audience, you also need to consider differences between novice and experienced users and include both types of users in your test.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

[1] The GVU was established in 1991 at Georgia Tech. You can find their recent research at www.gvu.gatech.edu/gvu/.

Step 2 – Define the Most Commonly Used Functions

In a typical software application, people tend to use a small number of functions most of the time. Although many functions are available and are paid for, people fall into patterns of how they use the application. For example, a Microsoft Word user may likely use tables, spell check, grammar check, formatting and help, while not using features such as macros and mail merge. This is not to say that usability is not important in the lesser-used functions, but they are just not the highest priority for usability testing.

In planning the usability test, list the functions you want to have tested. These functions should be described in terms of what the user should do in performing them. For example:

Function to be Tested & Task

Search for products –

- Try to find specific products by entering a description in the search criteria.
- Try to find specific products by browsing the online catalog.

As contrasted to other forms of functional testing, test scripts and test cases will not give the kind of information you need to evaluate usability. You want to avoid coaching and detailed descriptions of how to use the functions. This is why simply identifying the functions along with a brief description of the task to be performed is adequate to guide the usability testers in performing the test.

Step 3 – Define the Evaluation Criteria

There have been a variety of methods used over the past several years to evaluate usability factors in software. These methods range from elaborate usability test labs to simple observation. My observation about the methods to evaluate usability is that the more elaborate methods can be helpful, but only add the remaining 20% of the information. You can often get the most essential and revealing usability information from direct observation of actual users along with a video to record the test for later reference. Unfortunately, some organizations have neglected usability testing altogether because they felt the expense and time were more than they could afford. This does not need to be the case at all. As we will see further in this process, you can do a lot of usability testing with a few resources!

How to observe usability testers

This is where the entire usability lab issue comes into play. My experience has been that you need a way to observe users that is:

- Non-intrusive

This is why some usability test labs have two-way mirrors to keep the observers from interaction with the testers. However, if the participants can control themselves to prevent interaction, you don't need the mirror in between.

- Gives a record of the test for future reference

This is where videotaping is very helpful. Some usability labs have multiple cameras positioned at various angles to capture hand actions, facial expressions, and utterances. Believe me, there are times you don't want to record the utterances. However, a problem with multiple cameras is that you also need a way to edit the multiple camera images together to make sense of what the tester is doing. This kind of set-up can get expensive, but is still nice to get that last percentage of information from the test. My experience is that a single video camera positioned at the correct angle gets most of the information you will need to evaluate the test.

How to measure test results

There are several ways to measure usability factors. A subjective way to measure usability is to have the testers rate each function on a scale of 1 to 5, where 1 is "very hard to use" and 5 is "very easy to use." Yes, this is a very subjective measure, but usability is very subjective as well.

Another measure of usability is to count how many distinct steps it takes a tester to perform a task. For example, to order a product, do they have to complete three web page forms or five? If I'm ordering a product for the first time from a company, I don't like to spend a lot of time filling out information for their demographic files. If a customer senses that the process is going to become involved, chances are they will click out and go someplace else online to shop.

Timings can also be used to measure usability. The shorter a process takes to complete, the easier it is to use, at least in theory. To use timings, you will need someone with a stopwatch to time the user, or you will need to go back through the video to capture the timings. To ask the user to keep their own timings adds to the complexity of the test and may skew your results by increasing the testers' frustration level.

Errors encountered during a process can also be a revealing measure of usability.

Step 4 – Design the Test

In designing the test, you need to know four things:

1. How to control the test.

   Control issues relate to things like when the test is to start and end, how the test will be measured and how the results will be assured as correct.

2. Input

   Inputs are the items that the usability testers will need to perform their tests. Examples include the list of functions to perform, sample test data, and any help functions associated with the web site.

3. Output

   Output describes the deliverables from the test, such as a video record of the test, a list of functions and timings, survey results, etc.

4. Procedure

   The procedure describes how the overall test process will be performed, not how each function will be tested, as in a typical functional test. The procedure should describe the basic steps to be performed in the test and who is responsible for performing them.

Step 5 – Build the Test Environment

We have discussed the pros and cons of the simple to elaborate usability test environments, so the main point in this step is to create the test environment that you have designed. Don't let this step intimidate you. In web usability testing, many people have found that all you need is a PC with internet connectivity (by the way, don't forget about testing at various connection speeds – it can be very revealing in terms of usability!) and a video camera. At the very minimum, just manually observing how a typical user performs functions on your web site can be very enlightening. However, the video camera gives you some hard evidence to show the web designers.

Before the test begins – and I mean less than one hour before – you need to validate one last time that the web site is accessible to the user from the test environment.

Step 6 – Perform the Test

Now that the test environment is in place, you are ready to start testing. Many people have found that having one user at a time perform testing is better than having two or more users in the room at the same time. This avoids the natural occurrence of users picking up on each other's frustrations.

The observer or facilitator needs to be in place and ready to record the test as it proceeds. The facilitator/observer also needs to be prepared to help resolve unrelated technical issues, such as dropped connections, system crashes, etc, depending on the level of assistance you want the user testers to have in resolving such problems.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

Step 7 – Record Test Results

As the test is proceeding, the observer/facilitator is keeping timings and making sure the user's actions are being videotaped. The user/tester may also be scoring each function based on the 1 to 5 usability scale.

Step 8 – Summarize and Report Test Results

After the test is completed by all users, the test analyst studies the results and creates a report of test findings. The report should be organized to highlight the most critical usability problems.

It's Not as Difficult and Involved as it Sounds !

Research by Jakob Nielson and Tom Landauer shows that most of the usability problems in a web application can be found by five testers or less (http://www.useit.com/alertbox/20000319.html). A lot can be learned by informally observing typical users and customers. The problem is that most web developers and designers don't get outside the organization to see how a typical customer uses the system.

Example: Go to three different office supply store web sites (e.g., officedepot.com, staples.com, and officemax.com) and try to find the same item, with price, description and availability.

You can also have usability testers compare the ease of use factors between your web site and your competitor's web site.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

You can also have a short survey question to ask users and customers to rate the usability of your web site. Keep in mind that "short" is the keyword and it means only one or two questions, not a multi-page survey.

One caveat to the 5-tester guideline is if you are capturing metrics, you will probably need more than 5 users to get an adequate sample size. Nielsen recommends about 20 testers when capturing metrics.

Tools

Video Camera

As we mentioned earlier, you don't need a TV studio to capture the users' actions. One video camera costing less than $500 can do the job. If you get into multiple cameras, you will also need a video editing system to relate the different camera views.

Capture Software (Camtasia, Lotus Screen Cam, etc.)

I have had a lot of success with capture software such as Lotus Screen Cam and Camtasia to record actions to show designers at a later time. The advantage of these tools is that they are inexpensive, can create digital video files that can be e-mailed or stored on CD-ROM, and the capture the exact screen image, as compared to capture/playback tools, which record keystrokes and actions on objects. The downside to using capture/playback for documenting a test is that you may not see the exact thing the user originally saw during the test. This is due to environmental differences between recording and replaying the test.

Checklists

Checklists are often overlooked as test tools because they are manual. However, checklists add a great deal of consistency to a process and are relatively inexpensive to develop. In usability testing, checklists can be used to provide a generic set of features to be tested, such as searches, help, and site maps.

Conclusion

Usability issues can make or break a web site. Unfortunately, companies often place usability testing at a low priority, which means it may or may not be performed. It is common to have a web site that is functionally correct, has good response time, and meets a customer need, yet is hard to use. Therefore, the customers will be more inclined to find a competing site that is easier to use.

The best time to perform usability testing is early in the development process. Finding defects late in the web development effort often requires more than just tweaking the web interface. In some cases, web sites have been delayed from implementation significantly due to last-minute re-design.

The good news is that basic usability testing is not difficult or expensive to perform. The real challenge in usability testing is not to discount the findings as “user errors,” but to carefully consider the findings as an opportunity for improvement.

Frequently Asked Questions

by Randy Rice, CQA, CSTE

Q: What exactly is UAT?

A: My definition of UAT is:

"User acceptance testing validates that the system support business or operation needs from the user perspective. User acceptance testing validates that the right system was built."

Basically, UAT validates that the system will support business processes, so really it becomes a business process validation. The concern is that the system was work as specified, but when introduced to the real world environment unanticipated problems may appear. Therefore, as I teach UAT, I teach a process that uses the business or operational processes instead of requirements as the "target" of the tests.

My attitude is that the users should design, perform, and evaluate UAT. Users will likely need training and facilitation to own the test, but it should be their test.

As far as environments go, a model office is effective for business users and a simulated business/operational environment is good for non-office applications. However, much depends on the application and its scope. If the test is for an entire system, you will need to have several users and workstations. If you are working with a standalone PC application, then one PC and one user may be enough. Now, here's another wrinkle - if you are testing peripheral devices or things like wireless apps, then you need to get the actual hardware that will be used in the production use of the system.

Q: What do you call the testing that is done at the very end? Unit testing is complete, integration testing is complete and system testing is complete. Now you go to the actual environment and test the system. You may have already tested it on a simulator, if available. Is this network integration testing? Is it interoperability testing? Is it verifying the interaction of related systems? Is there a name for this?

A: Actually, your question hits on one of those fine lines of what is considered good practice in testing. The reason the term is hard to find is that the commonly used best practice is to perform those final phases of testing, such as system, UAT and perhaps model office, then do a phased implementation or complete system implementation to go live. Most testing purists would say that all testing should be complete before implementation. That's the theory at least.

In real-world application you will see:

Parallel tests, in which both the old and new systems are in operation at the same time, then a total cutover occurs.

Pilot tests, or beta tests, in which there is a limited and controlled deployment to find configuration and installation problems.

One of my past clients had a test they called "System Delivery and Installation Testing" or something similar to that. This was in a client/server environment as validated that the installation of the hardware, network and software were all correctly installed before going live at a particular installation.

Interoperability and compatibility concerns are a primary reason why people perform testing during implementation as a safety net to find defects in an environment that would be nearly impossible to create in a test lab.

There is a saying among testers that says once the producer stops testing and delivers the system, the customers start testing as they use the system for real. In other words, the customers will find the defects the testers don't find!

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

Q: What is a definition for Test Strategy? Is it the same as test plan? Is it included in the test plan documentation?

A: A test strategy is the high-level description of what the test is to accomplish and outlines risks, scope, tools and overall approach. The "who, what, when, where and how" details of the test are written in the test plan. The test strategy is about 1 - 3 pages in length and forms the basis for the test plan, which is about 12 - 15 pages in length, depending on the scope of the project. The test plan is more concerned with logistical details that the test strategy. The test strategy is a document for communicating what the test is to accomplish and the general parameters of the test. It is often included in the test plan in the "approach" section, usually in an abbreviated format.

Q: Is there a method to forecast the number of bugs and severity levels expected to be found in an application if there are no statistics to compare it to? I realize that factors to consider are the Quality Practices in place at the time of development i.e.: How good were the Business Requirements, Functional specs, Code Reviews, Code Walkthroughs, Test Cases, How may CR's were generated against each etc. Is there a generic way to calculate this information?

A: You are correct in your thought that there are many variables which can greatly affect the defect levels.

There is at least one broad benchmark for defect levels without having to use a sizing measure, such as lines of code or function points and that is the defect removal efficiency metric.

Capers Jones has related the defect removal efficiency level to the CMM level. Defect removal efficiency is simply the count of all defects found in your testing divided by the defects found during the entire life of product (for whatever period of time you define that to be). So, if you find 90 defects and your customers find 10 more in the next x months, the defect removal efficiency would be 90%. Studies by Jones show that companies at SEI CMM level 1 have an average defect removal efficiency rate of 85%.

You might find the following links helpful:

http://web2.deskbook.osd.mil/reflib/DAF/035GZ/012/035GZ012DOC.HTM

http://www.apl.jhu.edu/Classes/605401/hausler/cybrbob.html

http://www.year2000.com/archive/proby2k.html

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

Stages of Process Maturity – Moving Up in Capability Maturity Model (CMM) Levels

by Carl Chandler

The Capability Maturity Model is a model for better, more predictable software. The stages of process maturity are intended to progressively take a company from the initial unpredictable and poorly controlled stage to the optimized stage focusing on process improvement.

There are five levels of software process maturity. Each maturity level is measured by an assessment performed by the Software Engineering Institute (SEI) or an independent organization. The maturity levels are intended to place the focus on the process instead of the product making it easier to achieve continuous improvement in predictability and quality.

Companies should expect to spend approximately 12 to 18 months or more in each of the maturity levels in order to apply the lessons and achieve the benefits of process maturity.

Based on SEI assessment results as of April 1996 only 0.4% of companies have successfully reached the 5th maturity level with 68.8% at the initial maturity level. Benefits start to increase at levels 4 and 5. At maturity level 4 the process is measured and controlled. Maturity level 5 is a constant improvement of processes.

With a goal to reduce the level of defects in software delivered to customers/users many companies require a minimum SEI level from their vendors. The Department of Defense has a minimum requirement of SEI Maturity Level 3 where processes are defined and placed in writing.

Rice Consulting Services' Course Offerings:

If you would like to learn more about the information covered in both Randy's and Carl's articles we here at Rice Consulting Services, Inc. offer two excellent courses that will enhance your companies software quality process.

Structured User Acceptance Testing – 3 days

This is a practical hands-on seminar to convey effective methods to plan and conduct user acceptance testing. This is one of the few courses available that teaches a non-technical and easily learned process for testing computer systems from a business process perspective. This course deals with testing issues from both the process and human perspectives. You will learn the terminology, the unique issues, and the process for performing user acceptance testing. As a result of attending this seminar, you should have a good working knowledge of what it takes to plan and conduct a very effective user acceptance test in your own organization.

Structured User Acceptance Testing will help you become more comfortable and confident in designing and performing a test that models how an organization will use a particular application to conduct business. You will emerge from this three-day session knowing how to develop test scenarios, test scripts and test cases. You will also have a working knowledge of how to coordinate all of the aspects of a user acceptance test into a smoothly flowing test.
Whether you are planning to test a vendor-developed or in-house developed applications, the process and techniques covered in this course can enable you to identify the most effective tests and maintain a high level of test coverage.
User acceptance testing does not need to be overwhelming and intimidating. Learn the issues and processes for effectively testing business processes by attending this hands-on course.

To learn more about this course visit our web site.

Building an Effective QA and Testing Process for On-Going Validation – 2 days

This course is designed to teach participants how to control the quality of existing systems during ongoing operations and maintenance. This session is appropriate for software developers, managers, quality assurance and testing personnel, and systems support professionals.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

The workshop contains two team-based exercises which focus on having the participants write processes for ongoing testing and change control using their own organization's unique attributes.

For more information on these courses or one of our many other offerings please contact Carl Chandler at (405) 414-6759.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

Funny Pics

March 2001 Issue:

- **Testing Middleware and Firmware**
  by Randy Rice, CQA, CSTE
- **Building a Testing Foundation**
  by Carl Chandler