USABILITY ENGINEERING

This section condenses the writings and teachings of a handful of usability experts, practitioners, and observers into some basic guidelines for incorporating usability engineering and conducting usability tests. These experts include, but are not limited to, Alison Head, Jakob Nielsen, Jeffrey Rubin, Mark Pearrow, and the remarkable team at the National Cancer Institute who created www.usability.gov. Much of their work and descriptions overlap.

www.usability.gov

Usability Goals

In a book that predated the Web by two years, human-computer interaction expert Paul Booth outlined the major goals of an interface (*An Introduction to Human-Computer Interaction,* 1989). Nielsen echoed and expanded these goals for the Web in numerous usability writings.

1) The service should be useful.

The interface or product must meet specific goals. When goals are not clearly stated, a computer-centered—as opposed to user-centered—design takes hold. To determine usefulness, a Web master or Web team should set measurable objectives for the site. Rather than simply providing a lot of content—or a lot of links to content—libraries should concentrate on linking the success of a site to the existing business models of a library. For instance, do the redesigned Web forms result in increased document delivery requests? Does the new online chat service result in more reference questions being answered? Do the online renewal forms mean that fewer patrons are coming to the library to renew materials?

Without measurable objectives for the site or service, its success can never be gauged.

2) The service should be easy to use.

Too much competition exists on the Web for an interface to be difficult to use. Surfers and researchers alike will abandon a complicated interface in search of a simple one, even if abandoning the interface means sacrificing features or accuracy. The Darwinistic nature of the Web is troubling to those who view the Web as a vehicle for open communication; nevertheless, whether altruistic or for-profit, an easy-to-use Web site will always outlive a hard-to-use one.

3) The service should be easy to learn.

A corollary to ease of use, learnability has two components: users must be able to use a service with a known level of competence (especially important for purposes of testing), and users must be able to come back to the service after long gaps of inactivity and not face difficulties.

4) The service should cause few errors.

Nothing is more frustrating than a system that does not work, so the likelihood of errors must be kept to a minimum. An important counterpart to real errors is the perception of errors. Libraries must address preventing actual and perceived errors amid the myriad Web services they support. The Web has bred a new type of user who assumes that any failure is system-related, which raises a topic that will be addressed

later—user expectations.

5) The service should be pleasant to use.

This goal will most likely be the toughest for libraries. In task-oriented interactions, the goal is always to get in and out as quickly as possible. As Pearrow says, the goal of any Web site should be "the successful completion of user errands, so that they can get back to having fun." (Pearrow, p. 7)

This author's professional experience with library Web services has covered a lot of ground, from Web catalog interfaces and course reserves, to Z39.50 protocol and laptop lending services, and everything in between. No service, however, has ever been as popular as patron self-services. The simple ability to renew a book online gets more mileage than almost any feature you can build into a Web catalog. This simple, task-oriented service saves the user time and gives him or her the instant gratification that the function should allow. If libraries should create or preserve one group of online services, patron self-services should win hands down.

Methodology, or The Usability Toolbox

The usability test represents just one part of the methodology underlying usability engineering. Other factors, often taken together to describe usability testing are described here. Although based primarily on the work of Jeffrey Rubin, this list combines general usability methodology with some specific instructions geared toward engineering better Web sites. The toolbox term is borrowed from Mark Pearrow and James Hom. The metaphor implies that any or all tools may be used on a project. This toolbox is not a sequential procedure for conducting usability engineering. On the other hand, determining usability does not mean picking only one method from the list below and applying it to every project. Picking the right tool for the job is essential.

Tool #1: Participatory Design

Participatory design brings developers, business representatives (in this case front-line librarians) and end-users together to design a solution. This method is distinctive in that the usability tool directly involves one or more members of the design team itself. In a library setting, designer involvement is likely to exist anyway, but designers should use caution when performing usability studies themselves. Feedback can prove useful in the beginning stages of a project, but the closer the participant is to the product, the less likely he or she can provide unbiased critical feedback. Designers in this position can even fear being too critical of peers with whom they must continue to work closely. Another important distinction of this tool that sets it apart from usual practice in libraries is early involvement of users; many library development strategies include various stakeholders, but users are rarely included in the process during the early stages. Libraries are more likely to pursue user input from focus groups.

Tool #2: The Focus Group

This is the method most often employed in the library community. Libraries know their users well, which is a distinct advantage over the small Web

startup trying to build a useful product and gain market share at the same time. (The flip side is that creative service and information professionals have easily lured library clientele to their products and services, since the demand for information was already well-established by the existence of libraries.) Focus groups involve small group discussions, moderated by a trained facilitator. For the most part, participants are all gathered in the same location.

Though focus groups might not raise awareness of actual user behavior, they can certainly determine users' attitudes and beliefs about a new product or service; focus groups can even be used to gauge reaction to a prototypical service. The danger of focus groups can be seen in discussions that are influenced by the group dynamics. Facilitators do not want focus group meetings to turn into two-hour gripe sessions.

Virtual meetings, though they can eliminate some of the personality issues of group meetings and allow greater anonymity, tend to represent what you often find in the high-tech world—complex solutions to simple problems. If the participants are chosen carefully (five to 10 is usually a good number) and the list of topics is chosen beforehand, a focus group can prove useful. Most importantly, the moderator must control the meeting; if possible, a skilled (and scripted) moderator should facilitate a productive focus group.

Focus group sessions can be conducted over and over again at little cost other than staff time. That cost can then be easily recovered by a more useful and user-centered service. Libraries should also consider some form of compensation for focus group participants: a free lunch, cool gadgets, or even a $5 copy card might suffice.

Tool #3: The Survey

Libraries also excel at implementing surveys. Unfortunately, the energy expended creating the data collected by a survey often seems greater than the benefit reaped from the data collection—either because the surveys are not carefully crafted to gather data that are useful, or because the data never comes back in a way that direct any change, but instead defend the status quo.

When the data are used, the results are good for generalizing small samples that can be applied to a larger population. Like the focus group, surveys do not add much to the evaluation of user behavior but can represent a useful forum for submitting experiences and attitudes. Good marketing of a survey helps the user to think of the survey as a wish list.

In the online environment, an online survey also seems fitting. In a recent survey of library vendors (conducted by this author), 30 paper copies of the survey were sent as hard copy and also made available online, the URL for the survey was provided on the print survey and also made available in a follow-up e-mail delivered one week after the printed survey was mailed. Though the overall response to the survey was only 30%, 100% of the respondents used the online version of the survey.

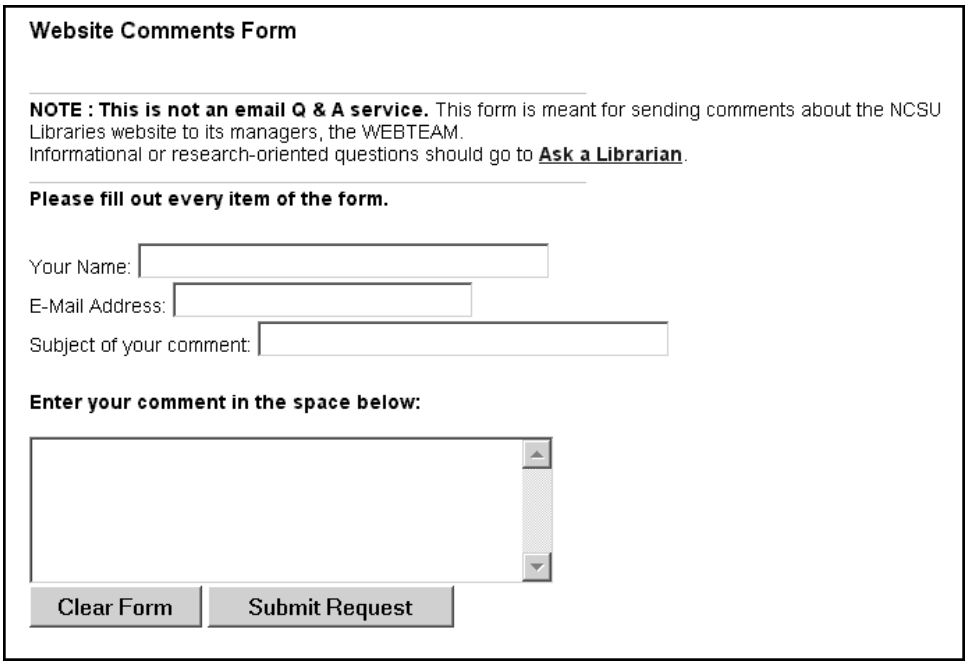
www.lib.ncsu.edu/systems/pace/vendorsurvey.html

An interesting trend has emerged with online surveys, with respect to their length. Readers in the IT field receive a barrage of solicitations for free magazines for which the only requirement is the occasional survey response. The designers of these surveys go to great lengths to hide the survey's own great length. The backlash against deceptive surveys has

already started since most people have no wish to start something when they cannot tell by looking at it how long it will take to complete.

Here are some simple rules for online surveys:

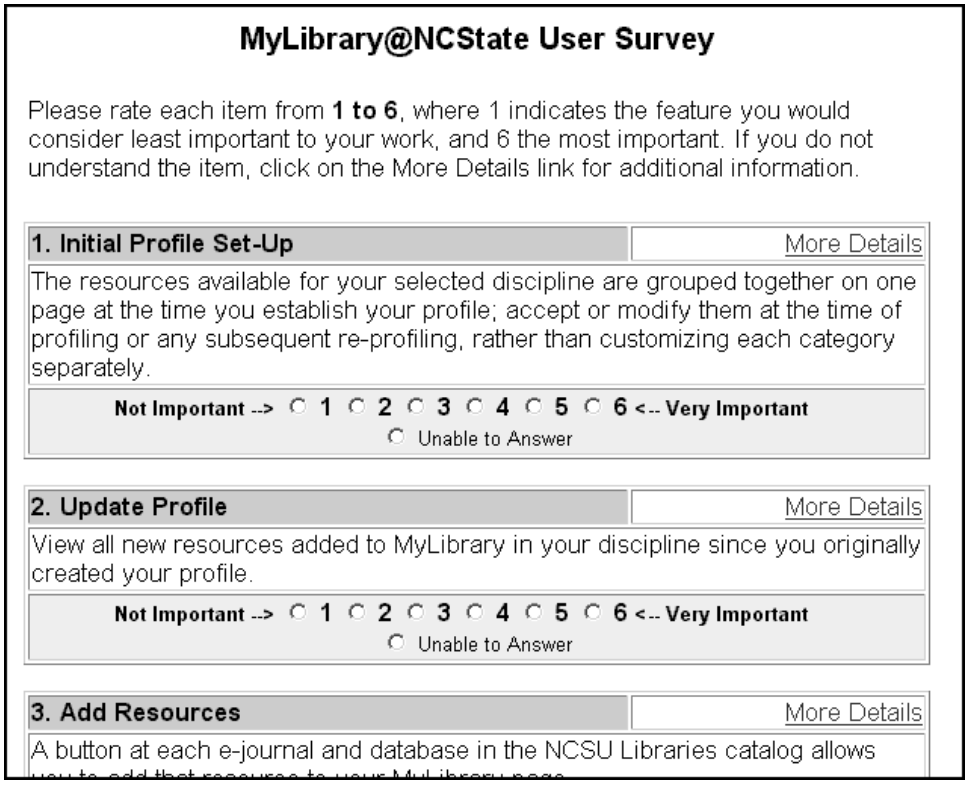
- Keep the survey simple. Ten questions to be completed in 10 minutes at the most.
- Use mostly closed-ended questions (For example, how many years have you used the Web?).
- Use open-ended questions only if you are prepared for the deeper analysis these responses require.
- Allow for anonymous responses but still try to determine what type of user is responding.
- Broadcast the availability of your survey. Don't expect respondents to stumble on it.
- Ask for permission to contact the respondent for follow-up and more in-depth questioning.



Here are some simple types of surveys:

Survey 1: Simple Comments Form. Completely open ended. From NCSU Libraries Web site: www.lib.ncsu.edu/Webteam-form.html

www.lib.ncsu.edu/
Webteam-form.html



MyLibrary@NCState User Survey

Please rate each item from **1 to 6**, where 1 indicates the feature you would consider least important to your work, and 6 the most important. If you do not understand the item, click on the More Details link for additional information.

1. Initial Profile Set-Up More Details

The resources available for your selected discipline are grouped together on one page at the time you establish your profile; accept or modify them at the time of profiling or any subsequent re-profiling, rather than customizing each category separately.

Not Important --> ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 <-- Very Important

○ Unable to Answer

2. Update Profile More Details

View all new resources added to MyLibrary in your discipline since you originally created your profile.

Not Important --> ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 <-- Very Important

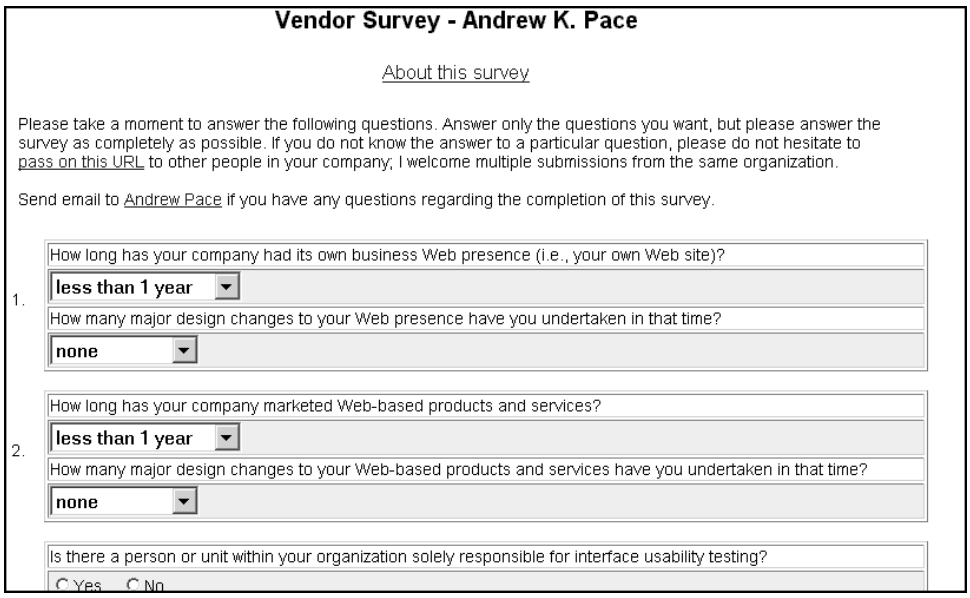
○ Unable to Answer

3. Add Resources More Details

A button at each e-journal and database in the NCSU Libraries catalog allows

Survey 2: Ordinal Data. MyLibrary User Survey To Gauge User Reaction to a Suite of Possible System Enhancements. From NCSU Libraries: MyLibrary Management Team, www.lib.ncsu.edu/mylibrarysurvey.html

www.lib.ncsu.edu/mylibrarysurvey.html



Vendor Survey - Andrew K. Pace

About this survey

Please take a moment to answer the following questions. Answer only the questions you want, but please answer the survey as completely as possible. If you do not know the answer to a particular question, please do not hesitate to pass on this URL to other people in your company; I welcome multiple submissions from the same organization.

Send email to Andrew Pace if you have any questions regarding the completion of this survey.

1. How long has your company had its own business Web presence (i.e., your own Web site)?
   less than 1 year
   How many major design changes to your Web presence have you undertaken in that time?
   none
2. How long has your company marketed Web-based products and services?
   less than 1 year
   How many major design changes to your Web-based products and services have you undertaken in that time?
   none

Is there a person or unit within your organization solely responsible for interface usability testing?

○ Yes ○ No

Survey 3: Mixed Opened and Closed Survey Questions. From Andrew K. Pace, Library Vendor Survey, www.lib.ncsu.edu/systems/pace/vendorsurvey.html

www.lib.ncsu.edu/systems/pace/vendorsurvey.html

Tool #4: The Individual Interview

The individual interview serves as a good follow-up to the survey. Like the survey, the interview tells you little about actual user behavior, but in the context of a completed survey, it allows for more in-depth follow-up. Interviewers should still rely on a script, although the interview should be thought of as a conversation. The interview can take place face-to-face, on the telephone, or even online via chat software.

Tool #5: The Contextual Interview

This type of interview is actually more akin to the usability test than to the individual interview. Much more natural than a formal usability test, the contextual interview takes place in the setting with which the user is familiar, such as an office or computer lab; the interviewer should observe and listen to actual user behaviors. The dialog can be informal, as long as purely qualitative results meet the needs of the usability encounter. The contextual interview sheds light on several aspects that might remain hidden in a formal usability test, like modem speeds, physical space limitations, browser preferences, and the like. While maintaining an informal air, interviewers should make careful notes either during the session or immediately after the interview.

Reference librarians know all about the contextual interview—it's their stock in trade. Pretend you have a medium-sized reference desk with 10 librarians, each working 10 hours per week on the desk. Assume a slow year and that they spend only 10% of their time working directly with users. Despite slow traffic at the reference desk, the otherwise busy library is closed only two weeks of the year. Still replete with print resources, the versatile reference staff goes online only half the time to help answer user queries. In one year, the reference staff has just conducted 250 hours of contextual interviews regarding online resource usage.

This example is not to meant to argue that reference departments operate in a vacuum—several members of the staff might be on the Web site or service design team. Although bits and pieces of the context gathered in everyday work might make their way into online library services, the interview process is altogether informal. A lack of hard data should not disqualify a volume business in contextual interviews. The information gathered should be presented formally and to maximize its potential for improving library Web services.

Tool #6: The Prototype and Walk-through

"Wanna see something cool?" Studying the increased usage of this phrase since the advent of the Web would prove interesting. Everyone wants to see something cool, and for those who make a living out of looking for, writing about, and showing off cool things, the Web offers a permanent escape from boredom. Unfortunately, the quick fix and instant gratification are byproducts of the Internet era.

When you find something cool or create something cool and want to share it, often the closest observer will do. The practiced conveyor of cool will seek out like-minded coolness experts; when self-congratulatory exercises prove minimally fulfilling, mutually enriching groups gather to bestow coolness on each other. Often, this grassroots method passes for prototyping a new service.

Though nothing is wrong with cool, it should not be used synonymously with good. A good idea or interesting service can be easily demonstrated to acquire consensus. One way to gain consensus is to attempt to replicate the idea on paper and test how users might perform with it. In the era of the WYSIWYG, mocking up simple HTML so users can provide feedback early in the design of a new service is often easier. Simple prototypes can even be used in conjunction with other tools in the usability toolbox, such as interviews and focus groups. Sometimes, early prototyping can kill a senseless project before it starts, or redirect it toward something more useful, either putting cool in context, or evolving something truly cool into something cool and, even better, useful.

Tool #7: The Card Sort

The card sort works well to determine the subject classification for new content or overall hierarchical organization. A more sophisticated version of prototyping, the card sort is especially useful for designing Web navigation and redesigning an entire Web site. Save this sort of test, for major reworking of the site; it would be difficult to apply findings in a card-sort test to a site that is already heavily used.

Here is a simple procedure taken from Mark Pearrow (Pearrow, pp. 66-67):

1. Get a set of plain and colored cards.
2. Put the name of each major category heading on the colored card.
3. Write the informational items on plain cards and give each one an ID number; give each card a short summary.
4. Pin the heading cards on a bulletin board, or spread them on a large flat surface.
5. Shuffle the plain cards and give them to the tester.
6. Have the tester put the informational cards in the best places.
7. Record the results using the IDs.

Testers should also consider offering a blank colored card to the testers so they can add new categories themselves. This option might prove useful for asking testers to categorize the varied content of a Web site.

The categorization issue is especially appropriate to libraries. Libraries have struggled for several years now in attempts to group electronic resources into categories that make sense to their users. In a card-sort test conducted by the University of Arizona (Dickstein and Mills), users were given a set of 82 titles to sort into logical categories. The design group was hoping to establish a group of about 10 subjects, but instead got from 13 to 37 groupings with agreement in the terminology on only a few. The librarians—in retrospect, admitting the librarian-knows-best attitude—ignored the findings and created 12 broad categories for the titles. After gauging user confusion in the first few months, the category offering was altered to more closely match the list created by the users, including redundant cross-referenced headings, and the users' own terminology.

For those who want the results of card-sorting and other metrics in a more electronic format, the Webmetrics guides and tools created by the National Institute of Standards and Technology (NIST) provide an even more technological approach to card-sorting and other usability engineering activities.

http://zing.ncsl.gov/WebTools/techn.html

http://zing.ncsl.nist.gov/WebTools

Tool #8: The Usability Audit

One way to seek quantifiable data on the usability of a Web site is though systematic evaluation. The first type of audit is simply expert evaluation. Many experts can evaluate the usability of a site simply by using it. Organizations might recruit or hire these experts to evaluate a given site or Web service. Librarians often assume everyone's opinion should be given equal weight; although this tendency is explained by human nature and kindness, it nevertheless defies logic. Library staff can separate novice opinion from expert opinion by the presence of suggestions for improvement; that is, any opinion of general inadequacy of a product or service must be accompanied by suggestions for making the service better for the criticism to be taken seriously.

More often than not, budget constraints prohibit the hiring of a professional to evaluate your site, but many in the field offer their opinions freely. The test for the organization then becomes measuring the opinions of others.

Another type of audit compares general standards. With this method, the design of the site is compared with a known body of standards either already established, or one created by the organization itself. Compliance with the laws of the Americans with Disabilities Act, for example, might necessitate comparing the HTML content of a site with the standards that remove barriers of access for people with disabilities. The W3C—the governing body of World Wide Web—establishes many of these standards, and several software packages and Web sites are available to test compliance; Bobby is probably the most popular. This topic will be addressed later.

www.w3c.org

www.cast.org/bobby

And finally, there are heuristics. Simply put, applying heuristics involves using general knowledge gained by experience, in layman's terms, a rule-of-thumb. A combination of the first two types of audits, heuristics for the Web involve the systematic inspection of an interface to determine if the site design complies with recognized usability. This explanation does not necessarily mean an expert must evaluate your site, but it might help to use existing heuristics developed by professionals. Jakob Nielsen's list of heuristics is probably the best known, and Mark Pearrow offers additional heuristics specifically geared toward the Web.

Heuristics is general knowledge gained by experience.

www.useit.com/papers/heuristics/heuristic_list.html

This list of 14 heuristics combines both lists and presents them in the context of library services.

1. Visible system status

A good system makes the user aware of what is going on; think, for instance, of the nearly ubiquitous use of "Your call is important to us…" recordings on business on-hold messages, especially those that tell you how much longer you have to wait. Unlike older systems that simply left you guessing as to what was going on, these messages inform the user periodically about system status. In an online environment, messages should come in a timely manner and be unambiguous. Searches that take a long time to run, for example, should notify the user; online request mechanisms should summarize the activity.

2. Match the system and the real world

Information should appear in a logical order and speak the language of the user. Charge and discharge, for example, are not part of the language spoken by people who want to check out and return books.

3. User control and freedom

A system should always provide an easy way out. More and more, it seems escaping a Web site is like trying to avoid a telemarketer. Javascript pop-ups and attempts to trap a user within your site practically guarantee no return visit.

4. Consistency of standards

Systems should avoid different words, situations, and actions that perform the same function or convey the same message. This wisdom is especially true when designing a navigational tool for a Web site.

5. Error prevention

If your online system cannot prevent a high rate of a certain type of error, then the system should either warn the user or be programmed to work around the common mistake. One of the most common examples involving library systems is the user who forgets to omit leading articles (such as: a, an, the) in phrase searches. Web catalog designers have three basic choices for handling this common error, each with varying levels of complexity: 1) try to instruct the user not to use leading articles before the search is performed, 2) have the system instruct the user after the fact, and offer an option to reconduct the search, or 3) have the system remove the offending articles before the search is performed.

6. Help users recognize, diagnose, and recover from errors

Error messages should express themselves in plain English, not codes. Although library staffs might like to think that a 404 error is impossible on their sites, customization of these messages is always a good idea. Brian Kelly of the University of Bath provides a nice overview of the 404 in the online journal *Ariadne*. If a certain type of error is frequent and unavoidable, do whatever is needed to help the user recover.

www.ariadne.ac.uk/issue20/404

7. Help and documentation

The best system runs day in and day out and is used by thousands without a shred of instructional documentation or help text. Computing Utopia aside, however, documentation should be easily accessible, simple to follow, and focused on the tasks that users wish to accomplish.

8. Recognition rather than recall

Examples abound in the real world to demonstrate that recognition works better than recall. Money is a good example. Although people might not be able to draw the front of a $1 bill with perfection, Americans recognize one when they see it. This heuristic is especially troubling with the Web. Vendor sites beyond a library's control change with the whims of new designers, the changing hands of corporate stock, or as a result of usability engineering. When does recognition come? Recognition comes in not having to recall the system information from one screen to the next, despite the overall design; consistent architecture, layout, navigation, and style allow the user to gain familiarity with the Web site as it is used. Recognizable sites do not require the user to learn something new from page to page. As a corollary, system instructions should always be easily retrievable. (A later section of this report discusses why Web designs should change often to keep pace with Web users.)

9. Flexibility and efficiency of use

The system should cater to both experienced and novice users. Help texts, for example, should be readily available to novices without distracting

experienced system users. Click-throughs (the study of how many clicks moves the user to the desired information) are the online equivalent of cutting to the chase and should be considered carefully.

10. Aesthetic and minimalist design

Do not make superfluous dialog compete with relevant information for the users' attention. Explanations of system downtime do this quite often. Some system administrators (or their administrators) cannot resist an urge to justify service unavailability in ways that are meaningless to users. For example:

> While we understand your need to find information in our catalog regardless of the time of day, certain data integrity requirements that allow us to work more efficiently necessitate that the online catalog be unavailable for about 3-4 hours while we perform upgrades and validate the stability of our system. As such, the online catalog will be unavailable for about 3-4 hours beginning at 2am, May 15, 2001. Please feel free to contact the library if you have any concerns regarding this downtime.

versus

> The online catalog will be unavailable tomorrow from 2am-6am while system maintenance is performed. We apologize for the inconvenience.

For noncontent Web site messages, less is always better.

11. Data chunking

Calling all collocation experts: like data should be grouped together. Take, for instance, a public library system with several meeting rooms or a campus with dozens of computer labs. You might expect each library or campus department to give information on these labs on their own segment of a Web site; chunking that data together, however, would be more useful to a user attempting to find out what rooms are available at any given time.

12. Journalistic style

Most people are familiar with the inverted pyramid style of writing. Like newspapers, all the important information must come up front because it's likely the reader won't get all the way through the article. This truth increases 10-fold for the Web. For the foreseeable future, reading on the Web will remain an uncomfortable exercise. Put the important information up front. Also like a newspaper, the important information is above the fold and scannable (or online, in the first screen of information).

13. Avoid gratuitous use of features

Creeping featuritis, if not deadly, is extremely detrimental to the development of Web services. Attempting to be all things to all people usually serves no one at all. Don't add technology for technology's sake. Like the Supreme Court's explanation of obscenity, use of gratuitous features is hard to define, but people know it when they see it.

14. Keep download and response times low

No one likes a slow site. No one likes a site that takes forever to load.

Heuristics shown in the 14-point checklist above are especially helpful in evaluating existing sites. They might inform a decision to embark on a new design or simply stay with the one you have. Once established, anyone can use them against any service.

How do you proceed with the heuristics? Make a bar graph that lists the heuristics you are interested in evaluating. Each participant records the violations found on the site. Record the severity of each violation. Gauge the severity by its frequency, impact, and persistence. The breakdown might look like this:

1 –little to no impact whatsoever

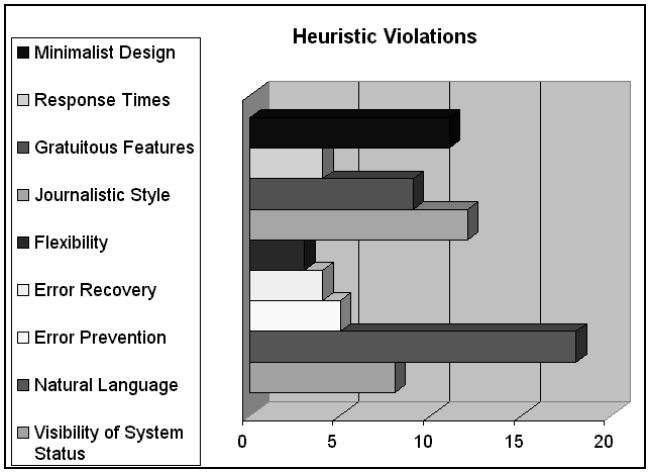
2 – cosmetic problem only

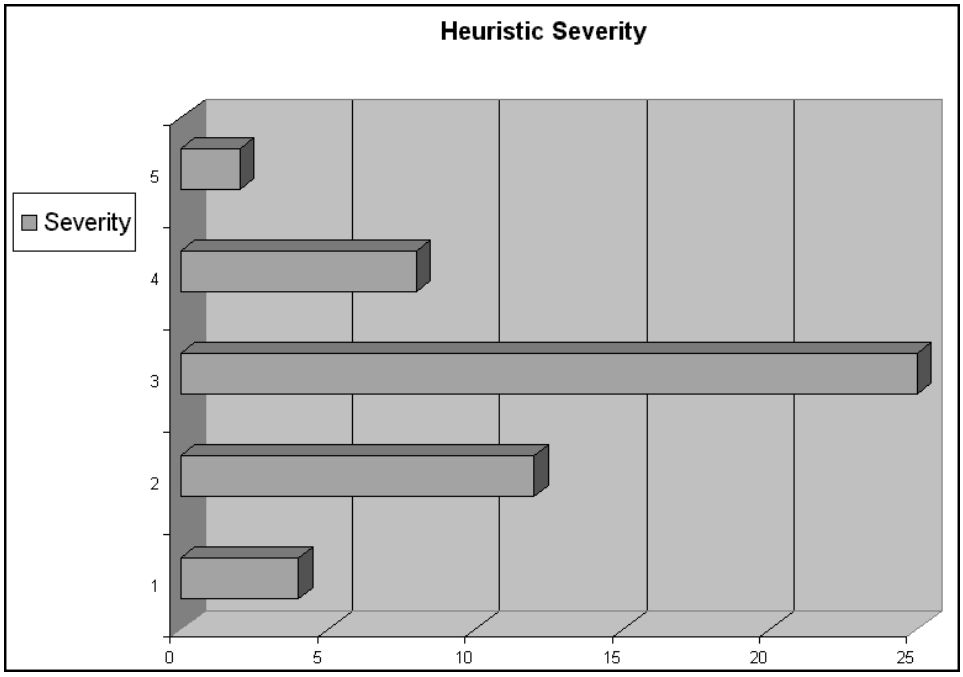
3 – minor violation

4 – major violation

5 – show-stopper

Using a sample of the heuristics listed above, your charts might look like this:





Recalling the 1-5 severity scale, this graph charts the severity of each heuristic violation, showing that the vast majority represents minor violations.

Although any number of evaluators can perform this heuristic evaluation, using multiple evaluators is always a good idea. The value of the data increases with the number of testers, but the law of diminishing returns kicks in at about five evaluators. This sort of evaluation can be conducted on existing sites and throughout the usability engineering procedure. Now that the heuristics are done, you also have a baseline with which to compare future tests. Keep these records not only for historical purposes but to compare the evolution of the basic functionality of your site.

www.useit.com/papers/heuristic

For even more details on heuristic evaluation, consult Jakob Nielsen's writings on the topic.

Tool #9: The Field Study

A field study puts the product or service in its natural environment. Think of a field study as a contextual interview combined with prototyping, usually in the late stages of development. The Web decreases the need for field studies, since the Web interface and many environmental variables can be reproduced elsewhere. More interesting, though, is the field study's evolution into the Beta test. (Interesting that these are not called omega tests, since they are usually a product's last stage before being foisted on the public.) A formalized visit, scripted questions, and quantitative measure of the field study is much more helpful than the try-this-and-tell-us-what-you-think approach.

Tool #10: The Usability Test

The most recognized and extremely misunderstood method of conducting usability engineering, the usability test is one of the most precise tools in the toolbox. Here's a recap to justify why testing is crucial:

- Testing ultimately saves the user time and saves the organization time and money.
- Testing counters the whims of designers.
- Testing is good public relations for the organization offering online services.
- Testing settles disagreements among design team members, whether stated or unstated.

Overview

www.usability.gov

The teams at www.usability.gov state the method most clearly: "Usability testing is an iterative process that involves testing the site and then using the test results to change the site to better meet users' needs. The best process is to try a prototype with a few users, fix it, and test again. Always keep in mind that this process is completely cyclical—the work is never done. Broadly, this is how to create a usability test:

1. Plan the scope: budget, participants, and location.
2. Create test scenarios.
3. Find test participants.
4. Conduct the tests.
5. Interpret the results.

This simplified overview does not address user needs analysis at the beginning of the design process. An organization at the testing stage of development should already have ascertained the need for the service. If that is not the case, be prepared for the usability test to show the service is neither helpful nor effective to the testers. Pearrow orders the development phase putting the usability test in context (Pearrow, pp.55-57):

Phase 1: User needs analysis

Phase 2: Operational and conceptual design

Phase 3: Development and testing

Phase 4: Validation (the flight-check)

Phase 5: Deployment

REPEAT—REPEAT—REPEAT

Unfortunately, usability testing usually works its way in after or during Phase 4, rather than before. More emphasis on Phase 3 helps the process immensely and makes for a smoother validation of your work.

Testing Scope

The major resource allocated for usability testing will always be time. Try not to think too much about glorious ergonomics testing, or human-computer interaction videos in which you have seen users' eye movements tracked and plotted in digital charts, or someone configuring an mouse-enabled stopwatch that records the milliseconds between a double-click. If this is the picture of usability testing that one webmaster or group of designers presents, testing may never get off the ground. Keep it simple, gain expertise, and build the organization's testing capabilities.

Testing Equipment

Some basic equipment is needed to conduct testing:

- A small, quiet room
- A digital stopwatch
- Two good chairs
- A workstation, browser, and Internet connection comparable with what will be typically used to access the new site or service

If you must have an extravagance, buy a video camera. For even more extravagance, buy two video cameras (one for the computer, and one for the face of the tester). Although there are ways to change the refresh rate of computer monitors so you don't see a wavy pattern when computers are filmed, an actual recorder for the screen itself might be easier.

Software that records the entire session eases the burden of lining up two videos to run side-by-side and creates a digital record of the test participants' experiences. These software packages range from inexpensive shareware to high-end sophisticated human-computer interaction packages. Often designed for security, these recorders can double as usability tools.

If the group does decide to record the sessions, make the camera on the participant as unobtrusive as possible. If the camera must be clearly visible

SpyWare

WinWhatWhere Investigator
www.winwhatwhere.com

STARR PC & Internet Monitor
www.iopus.com/starr.htm

PC Surveillance
www.thespystore.com/pcsurveillance.htm

UsabilityWare

ErgoLight
www.ergolight-sw.com

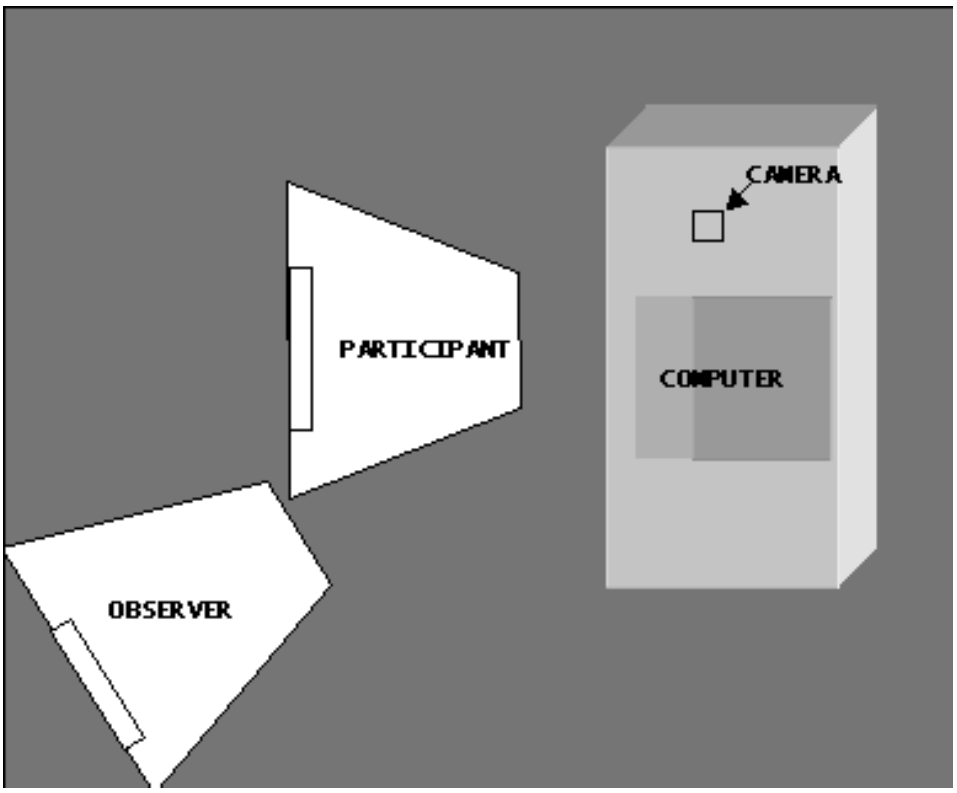
Human Factors International
www.humanfactors.com/library

Universal Usability
http://universalusability.org/resources/hardware.html

to the participant, then consider refraining from taping since users who are overly aware of being recorded can affect the test.

If the camera recording the participant is hidden to avoid interference, make sure that you receive the proper release forms for recording and let the participant know that the session is being recorded. (Note: Most colleges or universities have some form of human subject or talent release form. See Appenidix B for an example.)

The room designated for testing should always be reconfigurable, depending on the type of test being run and the number of people in the room. It is important that the room be comfortable and quiet. Also, the room should have no clocks. Like Las Vegas, the participant shouldn't keep close track of time because clock-watching affects performance. Finally, if you are lining up several participants, be sure to have a good waiting area for those who show up while others are conducting tests. Here is a simple room configuration.



View from above: sample usability lab with one participant, one staff observer, camera, and computer.

Staff Participants

Knowing that the following list of suggested participants might well be a single person, here is a list of the roles in a usability test group:

- **Test Administrator.** Responsible for greeting the participants, running the test, and overall test monitoring.
- **Data-logger.** One person to log whatever data is to be captured.
- **Timer.** Could be combined with the data-logger, but an essential part of the team.

- **Video operator.** Another candidate to be combined with any of the roles above; designate someone responsible for the recording of test sessions.
- **Technical expert.** This vital member of the team might be the person who created the site, wrote specific programming code for it, or served as the key technical advisor to the development of the service. The technical expert should not be the sole tester of the new service. Human nature disqualifies this person from objective participation in testing new designs, products, or services. That said, the technical expert is vital to the smooth execution of the test. The expert's insight into how the product works behind the screen serves the staff participants who conduct the test and adds perspective to what can and cannot be altered in the service.

Create Testing Scenarios

The test should begin with a purpose and problem statement. Simply stated, the purpose of the test is to determine whether the standards of usability have been met. The problem should state the specific things that the site, service, or product is trying to accomplish.

Think of the scenarios as a list of tasks you want the user to complete, and determine how you will measure success or failure for each task. The test can be as simple as finding a given page on a site, or as complicated as using every feature built into an online catalog. If you want to test an entire site, look closely at the existing site and identify the goal-oriented services that it provides (if you cannot find any, you might rethink the site altogether).

Like surveys, think about how many questions you want to ask, or how many tasks you want the participant to perform. As with the survey, make careful use of open-ended questions and the conversational component of your test; be sure that if asked for the information, the design team will be able to make use of the feedback.

Select the Participants

"Know your user" is the mantra of usability studies. Whether you pick two, five, or 10 users to conduct the site test, you must know the characteristics of those users to thoroughly evaluate test results. The sample that comes in the appendix of this report is not an exhaustive list of questions that you want to pose to testers. In addition to the personal data—age, education, Internet skills, and so on—pay close attention to the details of user constraints, habits, and preferences. Does the user prefer a Web-enabled phone or a 17-inch monitor? Is this a high-end user? How does the user currently complete similar tasks on the Web?

When searching for participants, librarians should consider the marketing involved in gathering volunteers. No one likes to think that he is being tested, and from grade school, people are conditioned with a negative reaction to the word. Replacing the word *test* is an exercise in semantics; any other term—usability experiment, usability exercise, usability survey—will do.

Once you have a pool of potential participants, randomly select from that pool to obtain desired test results. If asked the proper questions in the requests for personal data, a skewed pool of participants will be readily apparent in the usability report. For example, a new interface that requires

downloading several new plug-ins should not be tested solely by users with 5+ years experience on the Web. Pearrow issues a related warning about best-testers, those who are practiced enough to throw off the test results. His rule of thumb states that if someone sits down to test and starts reviewing the source code of your Web site, that person should be on the design team, not testing the product.

If you are redesigning a site or service, make sure to include a thorough test of the old site for control purposes. The control group participants should not be the same group that tests the new design. Even if the new design test results show high usability, without anything to compare these data with, the results have less meaning. Moreover, in the iterative process of testing, use different test groups in the before and after segments of the new design. Using the same group of participants to test the performance of the changes you made based on their feedback will not reap accurate data. Snake-oil design—in which the participant is miraculously cured of an ailment he suffered the week before—results in an inaccurate sense of security that the specific problem has been resolved. Use the data collected on potential participants to create a pool of similar participants for follow-up testing.

Conducting the Test

See Appendix A for details.

Quick review. You know why you are testing; you have a good testing area and staff participants lined up; you have well-thought-out test scenarios; you have test participants who meet the characterizations that have been predefined. It is time to conduct the test.

Prepare an orientation for the participants who will be testing the new service. Briefly introduce yourself and the team members. Remind the participant the Web site or service in question is being tested—this survey should not be construed as a test of user ability. Give the participant a general idea of what will be happening in the test, and remind the person to feel free to ask questions during the test, but mention that test monitors will not be permitted to help complete tasks. Encourage participants to speak their thoughts aloud during tasks, but speaking aloud can be unnatural and uncomfortable for some people. Ask participants if they have any questions before the study begins, and then be sure to complete any necessary paperwork before the actual testing of the service takes place.

Even if the test scenarios are well thought out, you cannot merely sit back and watch the test unfold. While the study is being conducted:

1. Never register approval or disapproval with the participants' comments or reactions. It is important to show no vested interest in how the participant interacts with the Web site being examined.
2. Avoid the temptation to rescue participants, which can prove especially hard for public service librarians. If the test monitor is part of the design team, keep in mind that interacting with the Web site or service is most likely second nature by the time testing starts. Unless the participant is clearly frustrated or uncomfortable, try to let the test run its course. If the participant is frustrated repeatedly and unexpectedly throughout the test, the frustration could be a sign that the participant did not meet the characterization criteria the testing team was seeking.
3. Let the user finish. Be sure the participant, not the test monitor, believes that he or she is finished with a particular task.

4. Probe without leading. Retrieving useful qualitative data means finding out how the user interacts with the new site or service in the context of completing specific tasks.
5. Stick to the game plan. If unexpected bugs or technical problems arise, suspend the test until problems can be resolved. Only suspend the test for bona fide usability problems, but things such as power failures, testing equipment problems, or network connectivity interruptions. Examining how the user performs with an unanticipated error, outside the scope of the service being tested, does not serve the usability test.
6. Relax. Impartial and solemn are not synonyms. Humor is good and helps to lighten what can seem like an unnatural setting for using the Web.

Debrief the Participant

Once the actual test is complete, have a general questionnaire or survey for the user to evaluate the test. This time should not be wasted, and one of the staff participants should review any written comments (if there are any) from the test to gather feedback while the experience is still fresh in the participant's mind. After users complete the post-test questionnaire, review the post-test for inconsistencies with test performance—for example, instances where a participant showed signs of faltering on a certain task but then noted that the task was simple to complete. Ask the tester to talk about these inconsistencies.

Similar to the focus on rescue avoidance during the test phase, the debriefing staff should concentrate on understanding problems, not on solving them. There will be plenty of time for analysis and re-testing after the user has gone, so avoid the temptation to convince the participant that all their issues will be resolved. Especially resist the temptation to start thinking immediately how to solve those problems until problems can be clearly evaluated.

Finally, be sure to leave the door open to future contact with each participant, both for follow-up on the initial test and for the possibility of participation in different future studies.

Tabulate and Summarize

Data gathered during the usability test should be transferred to spreadsheets or databases, and summaries of the test should be written soon as possible after the actual test. The difficult task of analysis rests in making the data mean something. Unfortunately for Web designers, few usability formulae look like this:

If Human-factor (x) / Computer-factor (y) < 42.6 milliseconds, then include the link in 14-pixel font without underlining.

Establishing benchmarks for what expectations you have of your test participants is key. If desired, the initial test can be run to establish those benchmarks. Subsequent tests (after making changes) will show changes in those statistics. A significant increase in usability indicates that the changes made were responsible and are not simply the result of random chance.

Below is a general approach to using statistical data in a usability report (see Rubin, pp.260-262 and Pearrow, pp.297-99).

Statistical Toolbox

There are several types of data that can be collected.

Nominal: named, but nonquantified data. More likely to occur in a survey or focus group, an example might be "List the five things you like most about the Web catalog."

Ordinal: lists the items that you ask to be ranked, for example 1=strongly disagree and 5=strongly agree. Unfortunately, this data lacks the size of gaps between ranked items.

Interval: ordinal data where the gap between measures is known, like a thermometer.

Ratio: equally sized increments that include zero. The most likely data from a usability test, so think of measurements like number of clicks, number of errors encountered, and so on.

To work with the numbers that are gathered, here are few tools:

Mean: "average" to the nonscientist, the mean is an easily digestible number for people who do not necessarily want to approach a report scientifically.

Median: literally, the middle result of your tabulations, the median can be a good indicator of the middle ground. Unlike the mean, it won't be skewed by either high or low numbers in the list of results.

Mode: the result that occurs the most often. Relatively meaningless for numerical results, the mode is more useful with nominal data; for instance, when a user selects a database to test searching, it might be interesting to note which database is chosen most often.

Standard Deviation: besides simply making your statistics look more professional by including something as simple as the standard deviation, the statistical measure does add meaning. Take, for example, measuring the amount of time it takes a group of participants to complete a given task. A standard deviation much lower than the mean time denotes that the participants are closely clustered, whereas a standard deviation close to the mean results from a broader distribution of results and could warrant a closer inspection of the results of a participant who needed more time to complete the task.

Standard Deviation: statisticians use the standard deviation to measure the variation in the distributions of statistical results. It is a tool that determines when widely distributed statistical measures should be a concern (such as why did this function take users anywhere from 3 to 60 seconds?).

The equation used to determine standard deviation is simple:

$$\frac{\Sigma x^2 - (\Sigma x)^2 / n}{n - 1}$$

Σx^2 = the sum of squares of each of the scores

Σx = the sum of all the scores

n = the total number of scores

Here is an example using mean, median, and standard deviation:

Time to find one book, one journal, and one electronic database title in the catalog

Session	x minutes	x minutes^2
1	2	4
2	4	16
3	9	81
4	14	196
5	6	36
Sum (Σ)	35	333

Mean = 7

Median = 6

Standard Deviation (SD) = 2.345

# of seconds to find online catalog from the homepage

Session	x seconds	x seconds^2
1	2	4
2	8	64
3	3.25	10.5625
4	3.25	10.5625
5	4	16
6	2.5	6.25
7	3.75	14.0625
Sum (Σ)	26.75	125.4375

Mean = 3.8214

Median = 3.25

Standard Deviation (SD) = 0.8

In the first example the SD is much closer to its mean than the second example, begging a closer look at some of the sessions in which users took much longer to complete their tasks. Data that result in an SD close to the mean on a consistent basis beg a much closer examination of the statistics being gathered and the success of the test altogether; either the participants represent a group that has too much difficulty with the service, or the service itself might need a complete redesign.

A word of caution: enough manipulation of numbers can result in any number of conclusions. Used consistently and with benchmarks to which results can be compared, statistics can add credence to any usability study.

To put all this into a real-life context, see Appendix A, which outlines a fictitious usability test report and some boilerplate forms and questionnaires that can be used for the process.

Acknowledgement: For the structure of this report: the work of the National Cancer Institute, Mark Pearrow, and Jeffrey Rubin. Karen Ciccone, head of the NCSU Natural Resources Library, authored the original version of the scenarios presented in the example.

Conclusion

Two things had to happen before usability engineering could really take off. First, someone had to prove that usability testing could save organizations money. That someone was Jakob Nielsen; his efforts in supporting the field should be appreciated by everyone who manages an online interface. Second, the scope and purpose of usability had to be acknowledged. The usability expert's focus is on the user, and not on some grand judgment of what service is useful.

Moreover, don't confuse the focus on the user with testing users—usability engineers conduct a test of the service and not the user. For those who end up conducting usability engineering on new Web services, whether through a formal effort or with some of the tools outlined here, the result is better service to your users.