The Usability Methods Toolbox Handbook

James Hom

1998

http://jthom.best.vwh.net/usability/usable.htm

Welcome to the Usability Methods Toolbox! I've attempted to compile information about almost all of the methods and techniques (well, ok, a lot of them) used in usability evaluation. I hope you find helpful information here, or at least a pointer to additional information that will help you find what you need.

The information in this site was compiled from numerous sources, but due credit must be given to my faculty advisor, Professor Lou Freund of the Industrial and Systems Engineering Department at San Jose State University, and my colleagues on the Cisco Systems Usability Team: Joelle Chapman, Kevin Flood, team leader Jim Nieters, and program manager Joei Cote'. Thanks, guys, for all your help.

● Check out the list of usability methods described in this site.

● Find out why this site is here.

● Learn more about usability methods from this list of usability sources. Many are available on the Web!

Table of Content

Contextual Inquiry

What is it?

Contextual inquiry is basically a structured field interviewing method, based on a few core principles that differentiate this method from plain, journalistic interviewing. Contextual inquiry is more a discovery process than an evaluative process; more like learning than testing.

Contextual inquiry is based on three core principles: that understanding the *context* in which a product is used (the work being performed) is essential for elegant design, that the user is a *partner* in the design process, and that the usability design process, including assessment methods like contextual inquiry and usability testing, must have a *focus*.

For example, suppose you need to assess the usability of a wrench for automotive repair. Using contextual inquiry, you'd visit mechanics at auto repair shops and see how they work. You'd take in not only physical arrangements such as the location of the tool chests, or cramped conditions inside engine compartments, but also environmental concerns, such as the level of cleanliness of their hands, or the noise level in the shop, or the tight schedules imposed by their bosses. All of these would help define a context for their work--and thus a context for the usage of your product, the wrench.

You'd also listen to their gripes about your product; how it slips out of their hands if they've been working on greasy stuff, how it gnaws the corners off stubborn bolts. You'd ask them what would make their jobs easier; what design changes would help them. They're a partner in the design process.

Of course, you'd conduct all this research centering on the one thing you're analyzing: the wrench. This focus is important--it sets the goals for the visit ("We need to know how they store their wrenches"). Once you're done with your site visit, you can assess from your notes whether you found out what you needed to know.

How do I do it?

Contextual inquiry follows many of the same process steps as field observations or interviews. Different considerations are kept in mind, however, with some portions of the process.

For example, interviewing during a contextual inquiry study usually does not include set, broadly worded questions. Instead, the partnership between the interviewer and interviewee is used to create a dialogue, one where the interviewer can not only determine the user's opinions and experiences, but also his or her motivations and context.

A lot of times, just having the interviewer around is going to make the interviewee a bit edgy. As the interviewer, you really need to be part of the user's world to be effective--sometimes, it takes a while before they're used to you hanging around. At that point, the job becomes much easier, since the users you interview will be more at ease with telling you what they *really* think about your product.

This usually means that this is a long-term study; you set up a relationship with the organization you're studying and agree on when you're going to visit, how often you'll be on site, and how long you'll be there each time. It's a lot like ethnographic studies where the ethnographer goes off to live in a particular culture for a year or two.

Figuring out who to interview is very important. Many times, the end user you're keeping in mind isn't the person that's going to be affected the most by your design or redesign. For example, when many corporate applications change or are upgraded, the person that is affected the most is the management information systems (MIS) person who has to go around and install the application on every computer in the building. Hanging around that person for a day will certainly give you an appreciation for ensuring that the installation process and interface is well designed.

Once you're done with the visit, assess whether you met your goals for the visit. Analyze your notes to determine questions for your next visit.

When should I use this technique?

Contextual inquiry is one of the best methods to use when you really need to understand the users' work context. Many times, the environment in which people work really influences how people use a product. It sounds like a cliche', but there really are people who print out their email and mark it up with comments before replying.

Also, this technique is great for finding out about work practices in domains that you know nothing about--whether it's lawyers looking up cases in a digital library, or roughnecks on an oil rig, or soldiers cooped up in a tank.

This technique is best used in the early stages of development, since a lot of the information you'll get is subjective--how people feel about their jobs, how work or information flows through the organization, etc.

Who can tell me more?

Click on any of the following links for more information:

Beyer, Hugh, "Getting Started with Contextual Techniques".

Beyer, Hugh, and Holtzblatt, Karen, "Apprenticing with the Customer: A Collaborative Approach to Requirements Definition," Communications of the ACM, May 1995.

Beyer, Hugh, and Holtzblatt, Karen, Contextual Design : A Customer-Centered Approach to Systems Designs, 1997, Morgan Kaufman Publishers, ISBN: 1558604111

Holtzblatt, Karen, and Beyer, Hugh, "Making Customer-Centered Design Work for Teams," Communications of the ACM, October 1993.

Holtzblatt, K., and Jones, S. "Contextual Inquiry: A Participatory Technique for System Design." in Schuler, D., and Namioka, A. (eds.) **Participatory Design: Principles and Practice**. Lawrence Earlbaum, Hillsdale, NJ. 1993: 180-193.

Ethnographic Study / Field Observation

What is it?

Observing users in the field is often the best way to determine their usability requirements. Traditional usability testing, while providing a laboratory environment that makes data collection and recording easy, also removes the user and the product from the context of the workplace. Sometimes, it's best to see exactly how things are done in the real world.

How do I do it?

You begin by arranging for a field visit with users. Choose a variety of users of your product, from different workplaces, industries, and backgrounds. You may have to arrange with your sales staff for contacts within your customer's organizations.
Use your time onsite wisely. You'll have just a few hours at the field site, so try to collect as much data as possible while you're there. You can analyze it later when you get back to the office.
Part of field observation is inquiry; that is, interviewing users about their jobs and the ways they use your product. Part is observation; watching people use your product in the way they normally would in the course of day-to-day life.
One way to ensure adequate data collection is to identify as many artifacts and outcroppings as possible:

- *Artifacts* are physical objects in use at a site (notebooks, forms, reports, spaces, walls)
- *Outcroppings* are noticable physical traits that mark or characterize the site (size of cubicles, size of whiteboards and what's written on them, uniforms written by certain castes of personnel). For example, in one hospital study, people who got to wear scrubs around the hospital had more status and influence than those who couldn't, either by management decree or by peer pressure.

Both of these terms come from anthropology--some mention the term *ethnographic observation*, which I take means "watching people."
Post-It notes can be both artifact and outcropping.
The layout of cubicles, and location of personnel (who sits next to the boss, who sits near the loading dock, etc) can be informative as well.
Someone you consult for advice or information is neither artifact nor outcropping, but can be characterized as part of a *relationship*.

How to Collect Artifacts and Data about Outcroppings

"Collecting artifacts and outcroppings" sounds like you're going on an archeological dig; in actuality, it's quite similar. In the same way an archeologist looks at the pottery of an ancient civilization to determine their nutritional intake, you can find objects during your field observation that will help identify how your users use your product. Perform the following steps:

- Identify the artifacts and outcroppings during interviewing/observation
- Collect and mark them onsite
- Take photos, get files on disk, ask for maps or layouts of physical objects
- You can do remote observation by sending a disposable camera out to a site, and have the people there take pictures of their environment. Once you get the pictures, discuss them over the phone with the people at the remote site.

Representing the Data

When using such data to form decisions or sway opinions about design alternatives, try some of the following representations:

- Show the artifact itself
- Show a photo of the artifact or outcropping
- Show a diagram of the artifact or outcropping
- Show a drawing of the object with the parts labled
- Show a drawing of the object before and after use
- Show repeated instances of the artifact or outcropping

Group Relationships

Group relationships can help identify process and information flows. They include organization, hierarchy, informal and formal links/interactions among groups, reporting relationships, etc.

Communication Patterns

Communication patterns show who talks to whom, and how often. For communication-intensive products, such as telephony, email, or advertising, this information is vital.

Inquiry

When asking people how they do things, or how they're supposed to do things, ask them, "Does that work?" "Do others do things differently?" "Why?"

When should I use this technique?

This technique is best used in the early stages of development, when you need to know more about the issues surrounding the use of a product rather than actual metrics. In the really early stages of development, when you just have an idea that you might need a product to satisfy this particular need, field observations help gather user requirements and issues for incorporation into preliminary designs.

Who can tell me more?

Click on any of the following links for more information:

Portions of this topic are from notes taken at Judy Ramey/Denise Carlevato/Erin Leanne Schulz's presentation at the 1996 STC Annual Conference.

Kane, Kate, "Anthropologists Go Native In The Corporate Village," October/November 1996, Fast Company magazine.

Nouveau-biz zine article on ethnographers finding work in corporations.

Macht, Joshua, "The New Market Research," July 1998, Inc. magazine.

Biz-zine article on using low-tech field studies instead of expensive focus groups for market and user research.

Tamakoshi, Laura, 1996, Field Methods, and Field Notes.

Online Web pages at http://www.truman.edu/academics/ss/faculty/tamakoshil/. Describes Tamakoshi's methods used during her anthropological research in Papua New Guinea. Geared toward anthropology folks, but still useful.

Wixon, D. , and Ramey, Judith (Eds.), 1996, Field Methods Casebook for Software Design, John Wiley & Sons, Inc., New York, NY. ISBN: 0-471-14967-5

Nice overview of ethnographic techniques used in software design--the only book on these techniques as used for software design that I've seen so far. Similar to the Wiklund book in that it discusses individual practitioners' experiences.

Interviews and Focus Groups

What is it?

Interviews and focus groups let you query users about their experiences and preferences with your product. Both are formal, structured events where you directly interact with users, asking them to voice their opinions and experiences regarding your product.

How do I do it?

You begin by formulating questions about your product based on the type of information you want to know. The sources listed at the end of this page provide more information on designing effective questions. Then, simply ask your users to answer these questions.
In interviews and focus groups, as opposed to surveys and questionaires, you're present to interact and facilitate discussion about the issues raised by your questions. With multiple users present, as with focus groups, the interaction among the users may raise additional issues, or identify common problems that many people experience.

When should I use this technique?

This technique can be used at any stage of development, depending on the questions that are asked. Often, interviews or focus groups are used after products are shipped to assess customer satisfaction with the product. More likely, though, interviews and focus groups are held at very early stages of development, when the product requirments are still not firm. Focus groups are then held to extract user requirements prior to initial design.

Who can tell me more?

Click on any of the following links for more information:

Greenbaum, Thomas L.,"Focus Groups: A Help or a Waste of Time?" from Tom's articles page at http://www.groupsplus.com/. Lots of other good focus group articles on that site.

Greenbaum also has a couple of articles on video focus groups: "Focus Group By Video - Next Trend Of The 90's" and "Is Your Moderator Ready for Videoconferencing?" on his site.

Greenbaum, Thomas L., The Handbook for Focus Group Research, 1997, Sage Pubns; ISBN: 0761912533

Nielsen, Jakob, "The Use and Misuse of Focus Groups"

Templeton, Jane F., The Focus Group : A Strategic Guide to Organizing, Conducting and Analyzing the Focus Group Interview, 1994, Probus Pub Co; ISBN: 1557385300

Maya Design and W3Focus do online focus groups: FAQ
Maya Design also does telephone focus groups.

Silverman, George, "How to Get Beneath the Surface in Focus Groups" from George's articles page at http://www.mnav.com/index.html. Lots of great focus group articles here, too.

Surveys

What is it?

Surveys are ad hoc interviews with users, where a set list of questions is asked and the users' responses recorded. Surveys differ from questionaires in that they are interactive interviews, although not structured like contextual inquiries nor formally scheduled and organized like focus groups.

How do I do it?

You begin by formulating questions about your product based on the type of information you want to know. The sources listed at the end of this page provide more information on designing effective questions. Then, simply ask your users to answer these questions.

When should I use this technique?

This technique can be used at any stage of development, depending on the questions that are asked in the survey. Often, surveys are used after products are shipped to assess customer satisfaction with the product. Such surveys often identify usability issues that should have been caught in-house before the product was released to the market.

Who can tell me more?

Click on any of the following links for more information:

Alreck, Pamela L., and Settle, Robert B., The Survey Research Handbook, 1994, Irwin Professional Publishing, Chicago, IL, ISBN: 0-256-10321-6 (ISBN is for cheaper paperbound 1995 version)

Salant, Priscilla, and Dillman, Don A., How to Conduct Your Own Survey, 1994, John Wiley & Sons, New York, NY, ISBN: 0471012734

Questionnaires

What is it?

Questionnaires are written lists of questions that you distribute to your users. Questionnaires differ from surveys in that they are written lists, not ad hoc interviews, and as such require more effort on the part of your users to fill out the questionaire and return it to you.

How do I do it?

You begin by formulating questions about your product based on the type of information you want to know. The sources listed at the end of this page provide more information on designing effective questions.

When should I use this technique?

This technique can be used at any stage of development, depending on the questions that are asked in the questionnaire. Often, questionnaires are used after products are shipped to assess customer satisfaction with the product. Such questionnaires often identify usability issues that should have been caught in-house before the product was released to the market.

Who can tell me more?

Click on any of the following links for more information:

Foddy, William, Constructing Questions for Interviews and Questionnaires: Theory and Practice in Social Research, Cambridge Univ Pr (Pap Txt); ISBN: 0521467330.

Kirakowski, Jurek, Questionnaires in Usability Engineering

Lessler, Judith L., Questionnaire Design in the Cognitive Research Laboratory, ISBN 0840604122.

Oppenheim, A. N., Questionnaire Design, Interviewing and Attitude Measurement, 1992, Pinter Pub Ltd; ISBN: 1855670445

Journaled Sessions

What is it?

Journaled sessions bridges usability inquiry, where you ask people about their experiences with a product, and usability testing, where you observe people experiencing the product's user interface.

Journaled sessions are often used as a remote inquiry method for software user interface evaluation. A disk is distributed to a number of test subjects containing a prototype of the software product, as well as additional code to capture (or journalize) the subjects' actions when using the prototype. Users perform several tasks with the prototype, much as in formal usability tests, and their actions are captured with the journalizing software. Upon completion of the series of tasks, the users return the disks to you for you to evaluate.

Because the journaling portion of the evaluation is largely automated, this approach to remote, hands-off inquiry is certainly more "usable" then self-reporting logging, where users are requested to write down their observations and comments and send them back to you.

How do I do it?

Journaled sessions allow you to perform usability evaluation across long distances and without much overhead. Once the code to journalize user's actions is in place, it is relatively inexpensive to distribute the test disk to a large number of participants.

Provide the users with prototype software, journaling software, and a script of test tasks for them to perform. You could make the script part of the journaling software, providing for a higher level of interaction from the user.

In addition to merely recording the user's cursor movements, mouse clicks, and other interface interactions, you can also provide dialog boxes in which the user types in comments or observations at various points in the execution of a task. With some thought, this method can approach the type of interactive inquiry promoted by contextual inquiry.

The main disadvantage of this technique is that there is no observer to "see" what the user is doing--the facial expressions of the user, or even spoken comments inadvertently expressed during difficult portions of the session.

Of course, provide a pre-paid mailing envelope for your evaluators to return their log.

When should I use this technique?

This technique is best used in the early stages of development--probably even pre-development, where the information you're attempting to gather is more preferencial than empirical. You'll want to ensure that your user pool is rather straightforward and honest, so you can assume their journaled sessions actually depict what they'd actually do with the product.

Who can tell me more?

Click on any of the following links for more information:

Castillo, José, Remote Usability Evaluation Home Page, 1998.

José has a ton of remote evaluation stuff on his page.

Nielsen, Jakob, Usability Engineering, 1993, Academic Press/AP Professional, Cambridge, MA ISBN 0-12-518406-9 (paper)

Clickometer is a shareware click counter.

Invisible Key Logger counts keystrokes.

Other software products that count clicks: RSI prevention software, automated software QA suites from companies like Mercury Interactive, Segue, Rational, etc.

Self-Reporting Logs

What is it?

Self-reporting logs are paper-and-pencil journals in which users are requested to log their actions and observations while interacting with a product. Like journaled sessions, this technique allows you to perform user evaluation at a distance. Unlike journaled sessions though, this technique requires much more work on the part of your subject user.

You'd use journaled sessions when you need detailed information from the remote tests; for example, the actual mouse movements or sequence of dialog boxes and menu items accessed by the user. Obviously, requesting the user to record all of their actions in a log, down to each individual click, is out of the question. (Although if you're lucky enough to get someone who's anal enough to do that, well, just think, is that guy representative of your user population? Good luck...)

Self-reporting logs, therefore, are best used when you don't have the time or resources to provide the interactive package required for journaled sessions, or when the level of detail provided by journaled sessions isn't needed. For example, you might want just general perceptions and observations from a broad section of users.

The main disadvantage of this technique is that there is no observer to "see" what the user is doing--the facial expressions of the user, or even spoken comments inadvertently expressed during difficult portions of the session.

How do I do it?

Provide users with a prototype of the product, a script of the tasks they are to perform with the product, and a journal in which to record their observations and tasks. It helps to provide the users with stopping points during the execution of their test tasks where they can write down their observations.

Of course, provide a pre-paid mailing envelope for your evaluators to return their log.

When should I use this technique?

This technique is best used in the early stages of development--probably even pre-development, where the information you're attempting to gather is more preferencial than empirical. You'll want to ensure that your user pool is rather straightforward and honest, so you can assume their journaled sessions actually depict what they'd actually do with the product.

Who can tell me more?

Click on any of the following links for more information:

Castillo, José, Remote Usability Evaluation Home Page, 1998.

José has a ton of remote evaluation stuff on his page.

Nielsen, Jakob, Usability Engineering, 1993, Academic Press/AP Professional, Cambridge, MA ISBN 0-12-518406-9 (paper)

Screen Snapshots

What is it?

Snapshots is a method where the user takes screen snapshots at different times during the execution of a task or series of tasks.

How do I do it?

Like most user testing, you provide the user with software and have him or her perform various user tasks with the software. In addition, you provide the user with a snapshot program and instructions for when and how to take the screen snapshots.

When should I use this technique?

This technique is best used in the early to middle stages of development, when you have some working software to be evaluated but are not to the point of requiring full testing. Snapshots are most often used in conjunction with other remote inquiry methods, such as journaled sessions or self-reporting logs. In journaled sessions, the journaling software usually tracks the user's interaction and can be thought of as a series of user-taken snapshots. With self-reporting logs, however, snapshots can provide a much-needed visual basis for interpreting the user's comments in their logs.

Who can tell me more?

Click on any of the following links for more information:

Castillo, José, Remote Usability Evaluation Home Page, 1998.

Nielsen, Jakob, Usability Engineering, 1993, Academic Press/AP Professional, Cambridge, MA ISBN 0-12-518406-9 (paper)

Heuristic Evaluation

What is it?

Heuristic evaluation is a variation of usability inspection where usability specialists judge whether each element of a user interface follows established usability principles. This method is the part of the so-called "discount usability engineering" method.
Basically, heuristic evaluation is a fancy name for having a bunch of experts scrutinize the interface and evaluate each element of the interface against a list of commonly accepted principles--heuristics. Early lists of heuristics were quite long, resulting in tedious evaluation sessions and tired experts. These long lists rather defeated the purpose of this method, which was to save time and money over testing. Nielsen distilled his list of heuristics down to ten that have served him and others well in evaluating designs.

How do I do it?

Get some experts

Gather a group of experts to perform the evaluation. Of course, the more experts you get to look at the interface, the more problems you'll find, but the cost goes up. In an analysis of six studies, Nielsen found that three to five evaluators detected most of the usability problems.
Some caveats should be made about selecting your experts. You'll want experts who are, well, experts, and know what they're doing. These folks should have a broad background in usability evaluation and human-computer interface (HCI) design. It might be hard to find an expert that knows the subject matter of the product ("domain knowledge") in addition to HCI expertise, but if you can, you'll get a lot out of that person. An example would be in evaluating do-it-yourself tax software--could you find an person who is an expert in HCI and tax accounting?

Experts evaluate on their own, then compare findings

Once you have your experts, send them off to perform their evaluation individually. They need to look at the interface on their own so that they don't get biased by their fellow evaluators. You'll need to provide them with the proper roles and scenarios to use so they'll have the right mindset and perspective when interacting with the product. If the users of the product will get job aids like manuals or online help, provide these to your experts as well. You want them to evaluate the total package.
The expert will go through the interface at least twice, looking at each element of the interface (for example, each menu item, button, control, affordance, whatever) and evaluating its design, location, implementation, etc. in regards to the list of heuristics.

Experts provide feedback

When each expert performs an evaluation, he or she can provide feedback in a number of ways. The following are a few of these feedback methods:
Structured Report: The expert writes up a formal report about his or her findings. This is probably the easiest to digest, since the evaluator will have compiled all of his or her notes and summarized things in the report, but it might delay the turnaround time.
Verbalized Findings: While evaluating the interface, the expert dictates his or her findings to another person. Although this adds the cost of another person, this can discover other problems that might be missed if the experts need to write everything down themselves. Plus, unstructured comments like "What the #{*&%+@$ was the designer thinking?!?" can get captured this way.
Categories: Before sending the experts off to do their evaluations, everybody agrees on specific categories of problems that they'll log. While this is really easy to analyze, it probably misses some problems that the other methods might find.
The experts usually then reconvene to discuss their individual findings. Most of the time, you'll get back a summary report of all the usability problems found, even if individual evaluators disagreed on whether a particular thing was a problem or not. Most reports provide the heuristic(s) that were violated by the problem, giving you an idea of how to fix it.

When should I use this technique?

Heuristic evaluation can be used at almost any time during the development cycle, although it's probably best suited to earlier stages, when you don't have anything firm enough to test. You can provide the experts with paper mockups, or even just design specifications, and still get a good amount of usability problems discovered before actual production work begins.

Who can tell me more?

Click on any of the following links for more information:

Instone, Keith, "Site Usability Evaluation".

Instone, Keith, "Usability Heuristics for the Web".

Nielsen, Jakob, "Heuristic Evaluation: How-To".

Nielsen has a bunch of other papers on his site too at http://www.useit.com/papers/heuristic/. Check them out.

Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Cognitive Walkthrough

What is it?

Cognitive walkthrough is a review technique where expert evaluators construct task scenarios from a specification or early prototype and then role play the part of a user working with that interface--"walking through" the interface. They act as if the interface was actually built and they (in the role of a typical user) was working through the tasks. Each step the user would take is scrutinized: impasses where the interface blocks the "user" from completing the task indicate that the interface is missing something. Convoluted, circuitous paths through function sequences indicate that the interface needs a new function that simplifies the task and collapses the function sequence.

How do I do it?

Begin by evaluating a system specification in terms of the tasks users will perform with that system. It helps to identify the user's goals and purpose for each task. For example, the interface for operating a car begins with the goals of opening the door, sitting down in the driver's seat with the controls easily accessible, and starting the car. And we're not even driving yet! This example shows the granularity that some walkthroughs attain. The goal of "opening the door" could be broken down into sub-goals: find the key, orient the key, unlock the door, open the door. Each of these goals requires cognitive (thinking) and physical actions. To open the door, do I orient my hand with the palm up or with the palm down? What affordances are provided for opening the door?
During the walkthrough, identify problems in attaining the goals. For example, some car doors accept keys only if they're oriented one way. Does this cause an unacceptable delay for the user? Since the sub-goal of opening the door is a prerequisite to operating the car, this might be a large issue.

When should I use this technique?

Cognitive walkthroughs are great for the early stages of development because they can be performed using just system specifications as a basis. Artists conceptions of what screens might look like can be used to give the walkthrough a more realistic bent.

Who can tell me more?

Check out any of the following sources for more information:

Rowley, David E., and Rhoades, David G. "The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure." CHI `92 Proceedings, (May 3-7, 1992): 389-395.

Spencer, Rick. " The streamlined cognitive walkthrough method." CHI 2000 Proceedings, (April 1 - 6, 2000): Pages 353-359.

Wharton, Cathleen, et. al., "The Cognitive Walkthrough Method: A Practictioner's Guide." in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Formal Usability Inspections

What is it?

Formal Usability Inspection takes the software inspection methodology and adapts it to usability evaluation. Software inspections, more commonly known as code inspections, started at IBM as a way to formalize the discovery and recording of software problems ("defects" in quality jargon, "bugs" in the vernacular). The technique also provided quantitative measurements that could be tracked using statistical process control methods. Code inspections were also adapted to check and track documentation defects, and usability defects were a logical next step.

Formal usability inspections include aspects of other inspection methods too. Heuristics are used to help non-usability professionals find usability defects. Inspectors walkthrough tasks with the user's goals and purpose in mind, similar to cognitive walkthroughs, although the emphasis is less on cognitive theory and more on encountering defects.

How do I do it?

This method formalizes the review of a specification or early prototype. The basic steps are to assemble a team of four to eight inspectors, assign each a special role in the context of the inspection, distribute the design documents to be inspected and instructions, have the inspectors go off on their own to do their inspection, and convene later in a formal inspection meeting. Defects found are assigned to responsible parties to be fixed, and the cycle continues.

Assemble the team. Pick a team of interested people, that is, people that have a stake in making the design more usable. This usually includes engineers from the design, quality assurance, documentation, training, and technical support groups. Each person brings a diverse viewpoint to look at the design, and the potential to discover usability defects is greater with a diverse team.

Assign roles. The formal usability inspection methodology borrows the inspection roles concept from code inspections. Each person on the team, besides having to inspect the design, has a role to play during the formal meeting. These roles are the following:

Moderator: Runs the meeting. Distributes and collects any materials needed. Schedules meetings, and coordinates defect assignment.

Owner: Designer of the product to be inspected. Usually the person to which defects are assigned. Fixes the defects.

Recorder (sometimes called Scribe): Logs defects during the formal meeting.

Inspectors: Everybody else. Inspects the design and reports any defects found. Everyone's an inspector regardless of their other role.

Distribute documents. For code inspections, this would be a code listing with line numbers plus instructions on what to look for--bad choice of syntax, variable problems, etc. For usability inspections, these include descriptions of the product, including screen mockups if any, user profiles, typical user tasks, heuristics to use, and a defect logging form.

Inspect the design. The inspectors work alone through the design and log the defects they find on the provided form. Having a form with an agreed-upon format for logging helps later during the formal meeting when the defects are discussed with the other inspectors. Each inspector assumes the role of a specific user from the user profile and walks through the tasks of a particular scenario. Prior to inspection, each inspector should review the heuristics and keep them in mind during their solo inspection sessions. Sometimes the form can be adapted to incorporate the heuristics as a checklist. Defects are logged according to the task the inspector was trying to execute and the location of the defect. With code inspections, the defect is located by line number--however, line numbers aren't usually present in interfaces. Defect location can be given as the screen and field or control name, or by the command and option attempted.

Hold the formal meeting. During the meeting, the moderator walks the team through each task/scenario as a group. Inspectors chime in at each step with the defects they found during their own inspection. Often, a lot of new defects are found as the inspectors discuss each defect--different aspects one inspector might not have thought of are brought up during the meeting. Everybody agrees on the recorder's logging of the defect--this formal log will be tracked later.

Inspectors might be tempted to think up solutions during the meeting, or the owner might take umbrage at the pronounced defects and protest each entry. These delays make the meeting run less smoothly and hurt the method's chance of success. Part of the mediator's role is to reduce these distractions so the defects can be aggregated and logged. There'll be plenty of time to fix them later.

Prioritize and fix the defects. Defects logged during the meeting are assigned to responsible persons to be fixed. The moderator often coordinates this effort, tracking fixed and open defects, and arranging solution-brainstorming meetings if necessary.

When should I use this technique?

Like other inspection methods, this technique is designed to reduce the time required to discover defects in a tight product design cycle. Since the inspectors can work with merely a specification or paper mockups, the technique lends itself well to early stages of development.

Who can tell me more?

Click on any of the following links for more information:

Kahn, Michael, and Prail, Amanda, "Formal Usability Inspections," in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Code Inspections

Freedman, Daniel, and Weinberg, Gerald M, 1990, Handbook of Walkthroughs, Inspections, and Technical Reviews : Evaluating Programs, Projects, and Products, Dorset House, ISBN: 0932633196

Gilb, Tom, Graham, Dorothy, and Finzi, Susannah, Software Inspection, 1993, Addison-Wesley Pub Co, ISBN: 0201631814

Wheeler, David A. (Ed.), Software Inspection : An Industry Best Practice, 1996, IEEE Computer Society, ISBN: 0818673400

Pluralistic Walkthroughs

What is it?

Pluralistic walkthroughs are meetings where users, developers, and usability professionals step through a task scenario, discussing and evaluating each element of interaction. Group walkthroughs have the advantage of providing a diverse range of skills and perspectives to bear on usability problems. As with any inspection, the more people looking for problems, the higher the probablility of finding problems. Also, the interaction between the team during the walkthrough helps to resolve usability issues faster.

How do I do it?

Most pluralistic evaluations involve a large group, with a usability professional as the session leader and facilitator. The group can consist of end users, product designers, documentation staff, and health/safety professionals in addition to usability staff. Everybody as a group discusses several scenarios derived from end user input. Based on their expertise as usability professionals, or experience as end users, usability problems are identified and classified for future action.

When should I use this technique?

This technique is best used in the early stages of development, as the feedback garnered from pluralistic walkthrough sessions is often in the form of user preferences and opinions.

Who can tell me more?

Click on any of the following links for more information:

Bias, Randolph G., "The Pluralistic Usability Walkthrough: Coordinated Emphathies," in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Feature Inspection

What is it?

Feature inspections analyze only the feature set of a product, usually given end user scenarios for the end result to be obtained from the use of the product. For example, a common user scenario for the use of a word processor is to produce a letter. The features that would be used include entering text, formatting text, spell-checking, saving the text to a file, and printing the letter. Each set of features used to produce the required output (a letter) is analyzed for its availability, understandability, and general usefulness.

How do I do it?

List the features in the product in the sequences they would be used to perform various tasks. Look at the accessibility of each feature in the context of the tasks. Can the user get to each feature without much trouble? Are the features well named and easily recognized?

One time-tested way to perform feature inspection is to have the documentation staff attempt to document each user scenario as procedures. Features that are hard to describe in the documentation are probably hard to find for the user in the first place.

When should I use this technique?

This technique is best used in the middle stages of development. At this point, the functions of the product--what the product is to be used for--are known. The features users will use to produce their desired output are known. The question that remains is how hard is it to use those features?

Who can tell me more?

Click on any of the following links for more information:

Bell, Univ. of Colorado

Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY.
ISBN 0-471-01877-5 (hardcover)

Consistency Inspections

What is it?

Consistency inspections ensure consistency across multiple products from the same development effort. For example, in a suite of office productivity applications, common functions should look and work the same whether the user is using the word processor, spreadsheet, presentation, or database program.

Consistency inspections begin with a usability professional analyzing the interfaces to all of the products and noting the various ways that each product implements a particular user interaction or function. An evaluation team then meets, and using the usability analysis as a basis, negotiates and decides on the one golden implementation for the usability attributes of each product.

How do I do it?

Form an inspection team, drawing members from each development team for all products covered in the inspection. These members should have the authority to negotiate for or against different design elements, and the power to change their product's design at the review meeting. Have a usability professional analyze each product as to its user interface, taking care to highlight areas that are vastly different from other products. This initial document will serve as the basis for the team's discussion during the meeting.

During the meeting, the team discusses the user interfaces to their products in terms of the usability professional's document, and for each element, comes to an agreement on what that element should look and work like in all of the products. This agreement must be unanimous by every member of the team, and buy-in from each product's development team (buy-in for their representative's decisions) should be procured prior to the convening of the meeting.

Keep a running tally of the changes to be made and decisions agreed upon. Any issues that cannot be resolved quickly should be "parked" and discussed at a later, more focused meeting.

When should I use this technique?

This technique is best used in the early stages of development, when initial development work has not progressed to the point where products that require extensive changes to ensure consistency will not require total overhauls. The ideal time for consistency inspections are when design documents for each of the individual products are almost done, and before any actual work on building the products has commenced.

Who can tell me more?

Click on any of the following links for more information:

Wixon, Dennis, et. al., "Inspections and Design Reviews: Framework, History, and Reflection," in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Nielsen, Jakob, Usability Inspection Tutorial, 1995, CHI '95 Proceedings

Standards Inspections

What is it?

Standards inspections ensure compliance with industry standards. In such inspections, a usability professional with extensive knowledge of the standard analyzes the elements of the product for their use of the industry standard.
For example, software products designed for the Windows environment should have common elements, such as the same functions on the File menu, a Help menu, etc. Or, products designed to be marketed in a particular country may have to conform to that country's ergonomic standards. Many monitors or keyboards are restricted from some uses in certain European countries due to occupational safety and health standards in those countries.

How do I do it?

In most cases, the inspection must be performed by an expert in the standard. For conformance to an operating system environment, perhaps the inspection could be performed by a representative of the operating system company. For conformance to a national safety and health standard, the inspector would probably be a member of that country's standards organization.

When should I use this technique?

This technique is best used in the middle stages of development, as the actual design is being developed with the given standard in mind. For some products, the standards inspection will be the major milestone allowing the initial design to proceed into manufacturability analysis and further engineering work.

Who can tell me more?

Click on any of the following links for more information:

Wixon, Dennis, et. al., "Inspections and Design Reviews: Framework, History, and Reflection," in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Nielsen, Jakob, Usability Inspection Tutorial, 1995, CHI '95 Proceedings

Guideline Checklists

What is it?

Guidelines and checklists help ensure that usability principles will be considered in a design. Usually, checklists are used in conjunction with a usability inspection method--the checklist gives the inspectors a basis by which to compare the product.

How do I do it?

You begin by deciding upon the particular usability guideline set you'll use to judge the attributes and interaction methods of the product's interface. Many guidelines have been published and can be used as published, although you may want to tailor the guidelines to suit the exact issues faced by your product's user. Guideline lists are usually long and take a lot of time to go through--since you'll be checking the product's interface against the list for each attribute or part of the interface, a long list isn't as desirable.
Nielsen, in the book Usability Inspection Methods, lists a small set of usability guidelines as follows:

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

When should I use these?

Use guideline checklists when performing usability inspections, such as heuristic evaluations or consistency inspections.

Who can tell me more?

Click on any of the following links for more information:

Wixon, Dennis, et. al., "Inspections and Design Reviews: Framework, History, and Reflection," in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Nielsen, Jakob, Usability Inspection Tutorial, 1995, CHI '95 Proceedings

General Concepts of Usability Testing

What is it?

Usability testing is carrying out experiments to find out specific information about a design. Tests have their root in experimental psychology, which used to mean a reliance upon heavy-duty statistical analysis of data. Today, with more emphasis on the interpretation of the results rather than actual data-driven figures, you see less importance given to the hard numbers and more to the other things you find out during the test. For example, a lot of tests done today use the thinking-aloud protocol in conjunction with some sort of performance measurement. While performance measurement is still useful, information gathered from the thinking aloud protocol often makes its way into the product faster--it doesn't need to be aggregated and analyzed before you can act on it.

How do I do it?

The overall process is simple; get some users and find out how they work with the product. Usually you observe individual users performing specific tasks with the product. You collect data on how they're doing--for example, how long they take to perform the task, or how many errors they make. Then analyze the data from all your experiments to look for trends. This section, based on Rubin's Handbook of Usability Testing, breaks these phases out in more detail.

Determine what you're trying to find out

What do you want to know about your product? Start with an overall purpose for your investigation; maybe you want to find out why tech support's call rate has gone way up since the last release. Or, your market share has slipped and you wonder if the other guy's product sells better because it's more usable.
Distill this purpose down into a few objectives for your test. "How usable is the product?" is not a good objective. The objective has to be something you can test for, for example: Does the delay in waiting for the Java applet to load cause users to leave the site? How difficult is it for a novice to do their long-form taxes using this software? Does the online help system provide enough tax code information? Is that information in easy-to-understand language, not government jargon?
Design your test
Identify the users you'll test--you'll need this because you have to go out and find some users, and knowing if you need novices, or experts, male or female or both, young or old or both, is important. Who are the target users for your product? If you're testing fighter jet displays, you don't want a horde of high school kids running through your test scenarios. If you're testing soft drink machines, you do want high school kids in your user population, in addition to the fighter pilots. This user profile of your

product's users is important in developing your test design and choosing your sample subjects.

Determine the experimental design. The experimental design refers to how you'll order and run the experiments to eliminate non-interesting variables from the analysis. For example, suppose you're testing tax software. Do you want subjects that have done their taxes using your software before, thus already having knowledge about the product? Maybe you want to run two groups of users through--rank novices in one group, and semi-experienced folks in another. There's a lot of information on experiment design in the usability testing books. If you want even more information on experiment design, see the references in the statistics and test design section of the bibliography--the quality craze of the 80's gave rise to a lot of interesting test designs that might be applicable (especially the ones that reduce sample size) to your situation.

Develop the tasks that your users will perform during each experiment. Of course, these would be derived from tasks that the users normally perform when they're using the product. Specify what you need to setup the scenario: the machine or computer states, screens, documentation, and other job aids that must be present. Also, specify what signifies a completed task--for example, if the user successfuly saves the edited document, or completes the manufacturing operation with a finished, in-spec part.

Specify the test apparatus. In traditional scientific experimentation, for example, biological or chemical research, from which usability testing methodology is ultimately derived from, the test apparatus would be the lab glassware, bunsen burners, flasks, etc. and other devices used in the course of the experiment. For usability testing, this is the computer and its software, or the mockup of the manufacturing workstation, or the prototype dashboard of a car.

The test apparatus can also include devices used in the running of the test, like video cameras to record the user's actions, scan converters to record the on-screen action, audio recorders to record verbal protocols, one-way mirrors to help the experimenter stay out of the subject's way, and so on. A lot of importance is placed on these items in regards to usability testing, but it really doesn't have to be that way. Even with simple home camcorders, or no video recording at all, you can find out a lot of useful information.

Identify the required personnel. You'll need at least one experimenter to run the test, from greeting the subject to explaining the test sequence to working with the subject during each task. You might also want to enlist an observer or two to reduce the data logging load on the experimenter.

Get some users

Assemble a list of users from which you'll draw a sample population of test subjects. There's so much written about picking subjects that it would be hard to list all the issues, but here's a few guidelines. You'll need enough users to fill your sample population with the correct mix of experience, skills, and demographic characteristics--otherwise, other factors might creep into your experimental design and influence the data. The user profile you determined during your experimental design will help you identify the exemplar user for your product. For example, a fighter jet display might be "used" by not only the pilots themselves, but also maintenance workers, installers, diagnostic testers, and instructors. However, for the purposes of what you're trying to find out ("Does the radar display of surrounding aircraft allow the user to avoid collisions during inverted flight?") you might be concerned only with one segment of the overall user population--the pilots.

Even if you've narrowed the user population down to a single profile, for example, "male or female fighter pilots with 20/20 vision between the ages of 22 and 35, with at least a bachelor's degree or equivalent," you'll still need to gather more information about them. How much experience with this type of display does each user have? Are they used to the old-fashioned mechanical gauges or do they prefer high-tech computerized displays? Are they colorblind? Which eye is dominant? You could go on and on, but the more knowledge you have about your sample subjects, the less you can be surprised by weird factors that skew your experimental data.

How do you find all these users? Well, by any means possible. Recruit from fellow employees and the family and friends of employees. Enlist temporary employment agencies and market research firms to get people (you might need to pay for them, but you'll probably have an easier time sorting through their characteristics). Get customers from Tech Support's call logs, or from Sales' lead lists. Offer free food at college campuses. Put out an ad on the Web, or in newspapers. Contact user groups and industry organizations. Consider other populations, like retirees who might have more spare time. Invite schools to send students over for a fieldtrip.

You might have problems finding particular user populations. If you need to test fighter pilots, can you get enough from each branch of the military to cover their specific biases? If you're testing an executive information system (EIS), can you procure enough executive-level people to test against, given their hectic schedules?

Setup the test

Prepare the test apparatus. The test apparatus for a usability test includes the computer, and its software for a software test, or the machine or mockup for a hardware test. Some tests are run with prototypes; if so, ensure that the states to be encountered during each task scenario will be available in the prototype. Also include the materials that will be provided to the subject and to the experimenter during the test. The subject usually gets a list of the tasks to perform. Often, the steps to perform each task are intentionally omitted if you want to determine the discoverability of certain command sequences. The experimenter usually gets a basic script of how to present the tasks to the subject user, and a form to help log observations during the test. Sometimes an agreed-upon shorthand for noting test events is useful in the rush to capture everything that's going on.

Prepare the test sample. The sample is the group of subjects you'll run through the test. How many do you need? Most common guidelines recommend at least four to five participants to find the majority of usability problems. Pick your sample based on your objectives and user profiles, and their availability on your test dates.

Run the test

Prepare the subject for the test. Since most people are uncomfortable when they're put into a laboratory and asked to perform tasks while being timed and their mistakes are logged for analysis, it's crucial to set the subject at ease. Explain that the user can stop the test at any time, go use the bathroom, or take a break if he or she needs to. Emphasize that you're testing the product, not the user, and that they need not feel pressured by the test. Thank the user for participating.

Most tests have each subject sign nondisclosure agreements and recording consent forms prior to the test. As a part of this filling-out-paper step, you can have the user complete a pre-test questionaire to identify domain knowledge or attitudes, or get more information about the user's characteristics.

Run the subject through the tasks and collect data. The typical test consists of a subject at a workstation, performing written tasks while the experimenter observes the user and asks questions or provides prompts if necessary.

Tests that are looking for primarily preferential or conceptual data (through thinking aloud, for example) can have a fairly large amount of interaction between the experimenter and the subject. For tests where you're trying to find out empirical data, like error rates, you'll want to reduce the interaction until it's a minimal influence upon the subject.

Let the subject work through the tasks without much interference. It will be tough to watch them struggle through tough parts, but it's better to learn from their struggling in the lab rather than have them struggle once they've paid for your product and brought it home. Of course, if a user really gets stuck to the point of tears or leaving the lab, assist them with getting through the immediate problem or simply move on to another task.

Even if you're not using a thinking-aloud protocol, you might want to ask the subject questions at different times during the test if you feel you'll learn a lot more about why the subject did something a certain way.

Debrief the user

Discuss the test with the user. After the tasks are complete, and the test is over, chat with the subject about the test. Go over events that happened during the test to gather more information about what the subject was thinking at that time. One way to review the events is to recall the event and discuss it with the subject, or to simply ask the subject which events were the most noteworthy.

Thank the user for participating. Remember, the subjects are here doing you a big favor, and it's important to let them know you appreciate them. Most labs provide a small gift for the subject: a coffee mug, or t-shirt, or free software, after the test. Many times, you'll want to draw from your pool of previous subjects for a future test, so it's important to keep them happy about participating.

Analyze your data

Find the big problems first. Identifying the big problems is probably easiest since they would be evident through observation notes. If every subject had a problem with a particular menu item, obviously that item's design needs to be revisited.

Summarize the performance data you've collected. Performance data like error rates and task durations is evaluated by performing statistical analysis on the data set. Most analysis consists of figuring the mean and standard deviation, and checking the data for validity. Does the data indicate any trends? Were particular parts of the product more difficult?

Summarize the preference data you've collected. By observing the user's actions, and recording the user's opinions, either during the test using a thinking-aloud protocol or asking questions, or before and after the test in the questionnaires, you have amassed a large set of preference data. Most questionaire designs allow you to quantify opinions using numerical scales, and the quantitative data found thusly can be analyzed using statistics much as the raw performance data. You can also summarize this data by selecting quotes from the subjects to highlight in the report as sound-bites.

When should I use this technique?

Usability testing is used throughout the product development lifecycle. In early stages of product development, testing the previous version or competitors' products gives the design team benchmarks to shoot for in the design. In middle stages of development, testing validates the design and provides feedback with which to refine the design. At the later stages, testing ensures that the product meets the design objectives.

Well, times do change. I got this email back in March of 1998, in regards to my citing Hagan Heller's article on low-cost usability testing from 1994:

You should note in your bibliography that while it has lots of good ideas, much of the information in it is significantly out of date. Some examples:

* Scan converters are not nearly as expensive, and are quite portable nowadays. We have one that does up to 800x600 resolution on both WinPC's and Macs that is in the $2000 range, which is about an order of magnitude less than she cites.

* Excellent portable labs with built in video and audio mixing capabilities and decent video editing are available in the $15,000 to $30,000 range. They can be set up in under an hour at any site (well, depending on how many flights of stairs there are) and can do titles and other effects when hooked up to a computer for editing. They DON'T require a Video Toaster, unless that's what's under the hood of the mix board in my lab now.

However, their note about how tripods are good and you need one is dead-on.

Thanks to Merryl Gross for the info. Your note does really ring true. We cobbled together the lab at Cisco on a really, really low budget. It was as if the "Tightwad Gazette" lady decided to construct a usability lab--scrounged desks and chairs, telephones set on "conference call" as our intercom, etc. Ah, those good old days...

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ
ISBN 0-89391-991-8 (paper)

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Additional Information

Bell, Brigham, Rieman, John, and Lewis, Clayton. "Usability Testing of a Graphical Programming System: Things We Missed in a Programming Walkthrough." Communications of the ACM volume/number unknown (1991): 7-12

Chartier, Donald A. "Usability Labs: The Trojan Technology."

Cline, June A., Omanson, Richard C., and Marcotte, Donald A. "ThinkLink: An Evaluation of a Multimedia Interactive Learning Project."

Haigh, Ruth, and Rogers, Andrew. "Usability Solutions for a Personal Alarm Device." Ergonomics In Design (July 1994): 12-21

Heller, Hagan, and Ruberg, Alan. "Usability Studies on a Tight Budget." Design+Software: Newsletter of the ASD (1994)

Jordan, Patrick W., Thomas, Bruce, Weerdmeester, Bernard, (Eds.), Usability Evaluation in Industry, 1996, Taylor & Francis, Inc., London, UK. ISBN: 0-74-840460-0

Lund, Arnold M. "Ameritech's Usability Laboratory: From Prototype to Final Design."

Whiteside, John, Bennett, John, and Holtzblatt, Karen. "Usability Engineering: Our Experience and Evolution" from Handbook of Human-Computer Interaction, M. Helander (ed.). Elsevier Science Publishers B.V. (North Holland), 1988: 791-804.

Wiklund, Michael E., Usability in Practice, 1994, AP Professional, Cambridge, MA ISBN 0-12-751250-0

Yuschick, Matt, Schwab, Eileen, and Griffith, Laura. "ACNA--The Ameritech Customer Name and Address Service."

Usability Testing Techniques and Issues

Bailey, R. W., et. al. "Usability Testing versus Heuristic Evaluation: A Head-to-Head Comparison." Proceedings of the Human Factors Society 36th Annual Meeting, (1992): 409-413.

Dayton, Tom, et. al. "Skills Needed By User-Centered Design Practitioners in Real Software Development Enironments: Report on the CHI `92 Workshop." SIGCHI Bulletin v25 n3, (July 1993): 16-31.

Jeffries, R., et. al., "User Interface Evaluation in the Real World: A Comparison of Four Techniques." Reaching through Technology: Proceedings of the 1991 CHI Conference, New Orleans, April-May 1991, NY: Association for Computing Machinery (ACM), 119-124.

Virzi, Robert A. "Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough?" Human Factors, v34, n4 (1992): 457-468.

Thinking Aloud Protocol

What is it?

Thinking Aloud protocol is a popular technique used during usability testing. During the course of a test, where the participant is performing a task as part of a user scenario, you ask the participant to vocalize his or her thoughts, feelings, and opinions while interacting with the product.

How do I do it?

You begin by providing your participant with the product to be tested (or a prototype of its interface) and a scenario of tasks to perform. Ask participants to perform the tasks using the product, and explain what they're thinking about while working with the product's interface.

Thinking aloud allows you to understand how the user approaches the interface and what considerations the user keeps in mind when using the interface. If the user expresses that the sequence of steps dictated by the product to accomplish their task goal is different from what they expected, perhaps the interface is convoluted.

Although the main benefit of the thinking aloud protocol is a better understanding of the user's mental model and interaction with the product, you can gain other benefits as well. For example, the terminology the user uses to express an idea or function should be incorporated into the product design or at least its documentation.

When should I use this technique?

Use this technique in any stage of development. Thinking aloud is a cheap way of getting a lot of good qualitative feedback during testing.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ, ISBN 0-89391-991-8 (paper)

Ericsson, K. Anders and Simon, Herbert 1985. Protocol Analysis. MIT Press.

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Nielsen, Jakob, "Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier," online Web page at http://www.useit.com/papers/guerrilla_hci.html.

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Co-Discovery Method

What is it?

Co-discovery is a type of usability testing where two participants attempt to perform tasks together while being observed. The advantage of this method over the thinking aloud protocol is two-fold:

- in the workplace, most people have someone else available for help
- the interaction between the two participants can bring out more insights than a single participant vocalizing his or her thoughts

How do I do it?

As with the thinking aloud method, you begin by providing your participants with the product to be tested (or a prototype of its interface) and a scenario of tasks to perform. Ask the participants to perform the tasks using the product, and explain what they're thinking about while working with the product's interface. Have them help each other in the same manner they would if they were working together to accomplish a common goal using the product.

When should I use this technique?

This technique can be used during any phase of development. It is ideal for Computer-Supported Cooperative Work (CSCW) products, groupware, and other products designed to be used by workers in team environments.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Question-asking Protocol

What is it?

The question-asking protocol simply takes thinking aloud one step further in that instead of waiting for users to vocalize their thoughts, you prompt them by asking direct questions about the product. Their ability (or lack of) to answer your questions can help you see what parts of the product interface were obvious, and which were obtuse.

How do I do it?

As with the thinking aloud method, you begin by providing your participants with the product to be tested (or a prototype of its interface) and a scenario of tasks to perform. Ask the participants to perform the tasks using the product, and explain what they're thinking about while working with the product's interface. Also ask them pointed, direct questions about the product; for example, "How would you send the email message?" Their response, either in terms of the product being tested or in other products from their past experience, will provide insights into their mental model of the product.

When should I use this technique?

Use this technique during any phase of development.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ
ISBN 0-89391-991-8 (paper)

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Performance Measurement

What is it?

Some usability tests are targeted at determining hard, quantitative data. Most of the time this data is in the form of performance metrics--how long does it take to select a block of text with a mouse, touchpad, or trackball? How does the placement of the backspace key influence the error rate?
Often these metrics are used as goals during the design of a product. Goals can be stated as stipulations, for example, "Users shall be able to connect to the Internet without errors or having to call the toll-free number," or "75% of users shall be able to complete the basic task in less than one hour." These benchmarks are devised during initial usability testing, either of a previous release, or of a competitor product.

How do I do it?

You begin by following the basic usability test concepts of determining a purpose, identifying test objectives, designing the tests, and running the experiment. For performance metrics, though, consider the following additional issues:

Objectives must be quantifiable

As before, the test objectives have to be expressed in testable terms, but when measuring performance, they have to be quantifiable. For example, you could ask the question, "What's more efficient, keyboard shortcuts or toolbar buttons?" A question worded this way could be tested with two interfaces, one using keyboard shortcuts, and the other using buttons. You'd record the performance of each user by timing how long it took them to execute a number of commands, and log their error rates.

Experimental design is really important

Since the goal of a performance measurement test is to gather valid quantifiable data, your experimental design must be valid as well. Quantitative tests assume that your change in the independent variable (for example, the presence of keyboard shortcuts or toolbar buttons) influences the dependent variable (time it takes to execute commands using one of the two options). This influence is called the experimental effect. However, if other factors are introduced into the design, the effect may be confounded, that is, not statistically valid due to tainting by the other factors. Your design must take into account possible confounding factors and eliminate possible sources of tainting.

Data doesn't tell the whole story

Testing solely for the purpose of procuring performance data doesn't seem to be as common as it used to be, for several reasons. Performance testing requires very rigorous test designs and extensive resources. Most companies don't have the time or money to do research of this kind. Also, the types of things tested are often at a very granular level. Does it really matter if it's half a second faster to use a keyboard shortcut than a toolbar button? Maybe if you're designing call center software, and amortized over thousands of operators across the country, saving each one half a second per call could save millions of dollars per year. But for most office productivity applications. half a second isn't really important.

When should I use this technique?

Performance measurement is used in initial stages of design to provide benchmarks for the design process. It's also used during the design cycle to measure the work done thus far against those benchmarks.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Eye Tracking

What is it?

Eye tracking allows testers to identify what participants look at during the course of a usability test. Eye tracking equipment uses several different technologies, including skin electrodes, marked contact lenses, image processing cameras, and reflector trackers. The last type is probably the most effective, as it does not require physical contact with the user's eye/eye socket. Instead, a beam of light is projected onto the eye; a sophisticated camera picks up the difference between the pupil refection and known reference points to determine what the user is looking at.

How do I do it?

Eye tracking involves sophisticated, usually expensive equipment. In most cases, unless the particular product you're testing requires eye tracking testing, it's best to find a usability lab that has already purchased an eye tracker and rent time at that lab for your testing.

When should I use this technique?

Use this technique when you absolutely have to identify what a person looks at during a usability test. For most products, regular inspection or testing methods will identify enough usability problems to the point that eye tracking isn't necessary.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Prototyping

What is it?

Prototyping models your final product and allows you to test attributes of the final product even if it's not ready yet. You simply test using your model.

How do I do it?

You begin by constructing a model of the final product--what it will look like, how it will work, and so on. With many user interfaces, this model can be as simple as paper-and-pencil drawings or as complex as actual working code. With hardware interfaces or workstations, this model can be cardboard or foamcore mockups. The closer your prototype is to the actual thing, the better your evaluation will be--however, you can get great results from very simple, "low-fidelity" prototypes.

When should I use this technique?

This technique can be used at any stage of development. As development progresses and the product is more complete, your prototype will encompass more and more of the product's final features. At some point, building additional prototypes will be less efficient than simply using early builds of the product.

There's a number of different terms you'll hear in conjunction with prototyping methods. The following is a sampling of some of these distinctions:

Rapid Prototyping: design methodology that quickly develops new designs, evaluates those designs, then "throws-away" the prototype when the next new design is developed along with a new prototype.

Reusable Prototyping: also known as Evolutionary Prototyping; effort used in constructing the prototype isn't wasted because parts (or all) of the prototype can be used to make the actual product. Mostly used in software development, although some hardware products can use the prototype as the basis for mold design in plastic manufacturing or auto body design.

Modular Prototyping: also known as Incremental Prototyping; new parts are added on as the design cycle progresses.

Horizontal Prototyping: prototype covers a large breadth of features and functions, but most aren't working. Great for testing breadth of scope but not actual use.

Vertical Prototyping: prototype covers only a narrow slice of features and functions that do work. Great for testing usage in a small portion of the product.

Low-fidelity Prototyping: prototype is implemented using paper and pencil, and thus mimics the function of the actual product, but doesn't look at all like the actual product. Great for testing on the cheap.

High-fidelity Prototyping: prototype is implemented to be as close to the actual design as possible in terms of look and feel, interaction, and timing.

Who can tell me more?

Click on any of the following links for more information:

Cooper, Alan, "The Perils of Prototyping," 1996

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Nielsen, Jakob, "Paper versus Computer Implementations as Mockup Scenarios for Heuristic Evaluation", Human-Computer Interaction-Interact `90, D. Diaper et. al. (ed.) Elsevier Science Publishers B.V. (North Holland), 1990: 315-320

Rettig, Marc, "Prototyping for Tiny Fingers (Everything I Need to Know About Prototyping, I Learned In Kindergarten)", Communications of the ACM, April 1994.

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Virzi, Robert A, Sokolov, Jeff, and Karis, Demetrios. "Usability Problem Identification Using Both Low- and High-Fidelity Prototypes," 1995: Obtained directly from the authors.

Low-Fidelity Prototyping

What is it?

Low-fidelity prototyping is a cheap way of providing prototypes to use in tests and participatory design sessions. "Low-fidelity" in this case means that the prototypes you use don't have to really look like the actual interface you're testing, as long as they "work" the same.
The idea is that you can get a lot of feedback about the interaction between the interface and the user by evaluating a low-fidelity prototype. Since low-fidelity prototypes are cheap, both in terms of money and time, you can afford to have more cycles of testing, more subjects, or more prototypes.

How do I do it?

Getting the prototype ready

The classic example of low-fidelity prototyping is using paper and pencil to mockup interface screens. These can be as simple as hand-drawn boxes with scribbled controls, or drawings done in a computer drawing program and printed out. Make one of each screen in the interface.
Also, make paper representations of menus, button groups, scroll bars, and the like. Post-It notes are good for this, as you can plunk them down on top of your "screen" as the user "clicks" on items, then rip them off as the state of the interface changes. Some people use transparencies for layering elements on top of one another. I think that's too much trouble. Besides, one of the cool things about low-fidelity prototyping is that you can invent new interface elements during the evaluation session, simply by drawing them onto a Post-it note. Transparencies are a bit messier and harder to work with during the evaluation.

Low-Fidelity Prototyping in action

Now comes time to use the prototype in an evaluation session. For example, in an informal thinking-aloud protocol usability test, you could use two evaluators, one to run the test, and one to manipulate the prototype. Evaluator Two in essence pretends to be the computer--she reacts to the user's input by switching paper mockups around or placing or removing elements on the current mockup.
Sometimes the action can get pretty hectic, especially if you have one of those expert users that loves to click everywhere. It helps to have a number of elements pre-drawn, and organized (I've used a chessboard to sort small Post-it note elements in a matrix) so you can easily grab an element when you need it.

When should I use this technique?

Like any prototyping method, use this technique when you don't have the "real" interface yet; namely in earlier stages of development. This technique is great when you don't have a lot of money or time to spend on prototyping, and you're more concerned with user feedback than actual performance data.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Nielsen, Jakob, "Paper versus Computer Implementations as Mockup Scenarios for Heuristic Evaluation", Human-Computer Interaction-Interact `90, D. Diaper et. al. (ed.) Elsevier Science Publishers B.V. (North Holland), 1990: 315-320

Rettig, Marc, "Prototyping for Tiny Fingers (Everything I Need to Know About Prototyping, I Learned In Kindergarten)", Communications of the ACM, April 1994.

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Snyder, Carolyn, "Using Paper Prototypes to Manage Risk," October 1996, Software Design and Publisher Magazine

Virzi, Robert A, Sokolov, Jeff, and Karis, Demetrios. "Usability Problem Identification Using Both Low- and High-Fidelity Prototypes," 1995: Obtained directly from the authors.

High-Fidelity Prototyping

What is it?

High-fidelity prototyping is a method where the prototype used for testing mimics the actual interface as closely as possible. Usually, in for software interfaces, another software tool is used to mockup the interface. This software tool accepts input from the keyboard or mouse like the actual interface would, and responds to those events in the same way (displaying a particular window or message, changing state, etc.) in the same way the actual interface would respond.

How do I do it?

Use a tool that allows you to mimic the look and behavior of the actual interface as much as possible. Many software prototypes are constructed using multimedia authoring tools, such as Macromedia Director or Authorware, Asymetrix Toolbook, or HyperCard. These tools allow you to construct on-screen interfaces that work exactly as the real interface would, including adding delays for slower machines or waits for "server" responses.

Web-based interfaces are probably best prototyped using HTML and Dynamic HTML. One nice thing about such prototypes is that you might get to reuse some of the code from the prototype in the actual interface.

Hardware prototypes can use foam core to provide a full-size physical representation of the product. If you are going to test extensively with the prototype, though, you might want to construct it out of sturdier materials, using a wood frame and plywood, for example. Some hardware (automobile dashboards, for example) are tested using mockups of sculpted clay over wood and wire frames.

When should I use this technique?

Like any prototyping method, use this technique when you don't have the "real" interface yet; namely in earlier stages of development. This technique is great when have a little more time or money to spend, and don't need to resort to low-fidelity prototyping. The better the prototype, the better the results. If the fidelity of the prototype is really good, you can use it in quantitative tests.

Who can tell me more?

Click on any of the following links for more information:

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Horizontal Prototyping

What is it?

Horizontal prototypes demonstrate a broad spectrum of the product's features, but without extensive functionality behind each function. Horizontal prototypes are often used for user preference testing of user interfaces, when the actual working functions haven't been implemented yet. Such prototypes allow evaluation of the interface design, feature placement and accessibility and the like, without requiring anything to actually work.

How do I do it?

Often, horizontal prototypes can be low fidelity prototypes consisting of little more than drawings and lists on paper, to computer mockups that look exactly like a finished product, except that none of the functions work. Realistic mockups can be developed in most graphical user interface (GUI) development environments such as Visual Basic or AppStudio, or in simple demo builders such as Demo It! or Hypercard.

When should I use this technique?

This technique is best used in the early stages of development, when work hasn't commenced on the actual functions of the product, but the feature set is known.

Who can tell me more?

Click on any of the following links for more information:

Nielsen, Jakob, Usability Engineering, 1993, Academic Press/AP Professional, Cambridge, MA ISBN 0-12-518406-9 (paper)

Vertical Prototyping

What is it?

Vertical prototypes demonstrate the exact functionality of a product but for only a small section of the entire product. For example, a vertical prototype of a word processor might demonstrate all of the spell-checking functions, but none of the formatting or text-entry functions. All of the functions in a vertical prototype mimic their real counterparts as much as possible.

How do I do it?

Since a vertical prototype needs to be practically fully functional (although just for a small portion of the product interface), perhaps the best way to obtain a vertical prototype is to use a fully functioning module of the product. For software programs that are written with a modular architecture, this can usually be done, although the interfaces to other modules won't work (ok in this case, as it's strictly the given section's functionality that will be tested or inspected, not other sections). For a car, it could be the seating and other interior furnishings that will be tested, while the drive train, body, sensors, and other components that aren't ready yet.

When should I use this technique?

Use this technique when the design for a particular section is rather complete and merits testing as a contiguous unit. Even though other portions of the product aren't ready for testing yet, you can determine problems with a particular portion while the others are still in an earlier phase of development.

Who can tell me more?

Click on any of the following links for more information:

Nielsen, Jakob, Usability Engineering, 1993, Academic Press/AP Professional, Cambridge, MA ISBN 0-12-518406-9 (paper)

Affinity Diagrams

What is it?

Affinity diagramming is a categorization method where users sort various concepts into several categories. This method is used by a team to organize a large amount of data according to the natural relationships between the items.
Basically, you write each concept on a Post-It note and tack them onto a wall. Team members move the notes to groups based on how they feel the concept belongs with other concepts.

How do I do it?

Perform the following steps:
1. **Form a team:** Gather a team of four to six people. There should be a good mix of experiences and perspectives, and they should approach the diagramming session with a creative, open mind.
2. **Describe the issue:** Have the team compose a statement about what they're trying to accomplish, or what the end result will be. This statement should be broad and neutral, yet clearly understood and agreed-upon by all members of the team. For example, suppose
3. **Generate Idea cards:** Brainstorm a list of ideas, then record each idea on a separate PostIt note. The cards should describe whole concepts, and not have just a single word on each one.
4. **Tack cards to wall**: Tack the notes to the wall in no particular order.
5. **Sort cards into groups:** Move the cards from into groupings. Just use your gut reactions for where the cards should go. No one speaks during this phase of the process, to make sure that no one influences someone else's decision. If you don't like where someone put a card, move it into a different group. Don't waste time disagreeing about where something should go.
6. **Create header cards:** Create header cards for each group. These header cards should concisely describe what each group represents. One way to think about this is to write a newspaper-style headline for each group. The header cards shouldn't be single-word titles, and should be able to stand alone and be meaningful without the contents of their groups.
Write sub-header cards for subgroups if necessary.
7. **Draw the Affinity Diagram:** Draw lines connecting the headers, sub-headers, and groups. Connect related groups with lines. The result looks a lot like a typical organization chart.

When should I use this technique?

Use this technique when you need to generate a large number of ideas or concepts and sort them into groups.

Who can tell me more?

Click on any of the following links for more information:

Brassard, Michael, ed. 1988. The Memory Jogger: A Pocket Guide of Tools for Continuous Improvement. Methuen, MA: Goal/QPC.

Blind voting

What is it?

Blind voting is a way for groups to vote on issues without the votes influencing others. Blind voting is often implemented as an electronic meeting system.

How do I do it?

Most blind voting implementations involve electronic meeting systems, specially equipped conference rooms where every seat has a terminal. These terminals are connected to a large display where each person's entries can be viewed. However, all entries are anonymous--no entry can be attributed to a particular person. These entries can be ideas or concepts, or votes.
In this way, meetings can't be taken over by overly strident or commanding people. Individuals who often are not very active in regular meetings can contribute even if they are normally shy.

When should I use this technique?

This technique is best used to ensure equality among team members in brainstorming and voting sessions.

Who can tell me more?

Click on any of the following links for more information:

Facilitate.com has a list of electronic meeting tips.

http://www.facilitate.com/Tips/
Ventana East is one vendor of blind voting systems.

Card Sorting

What is it?

Card sorting is a categorization method where users sort cards depicting various concepts into several categories.

How do I do it?

Start with a list of all the items you want sorted. Write down each item on a separate index card. Give your user(s) the stack of cards and have them divide the cards up into piles, telling them that the cards should be grouped the way they (the users) best see fit.

When should I use this technique?

This technique is best used in the early stages of development. For example, you might want to determine how users would group various functions. You'd use this information to place the functions on various menus of a user interface.

Who can tell me more?

Click on any of the following links for more information:

Brassard, Michael, ed. 1988. The Memory Jogger: A Pocket Guide of Tools for Continuous Improvement. Methuen, MA: Goal/QPC.

Education Evaluation

What is it?

A reader from Australia inquired about several methodologies that I had never heard of before. So, I went out looking for information about the following methods:

- stake model
- Stufflebeam (CIPP) model
- Scriven model
- Action research model

Here's what I found.

Stake, Stufflebeam, and Scriven, oh my!

These three models are often mentioned in the same breath; at least, I found them mentioned on the same web page. All are various methodologies used to evaluate education/training programs and curricula.
Stufflebeam's model is also known in some circles as the CIPP model, where CIPP is an acronym for Context, Input, Process, and Product.

Action Research

Action research is another methodology that comes from the world of education evaluation. It seems most similar to several participatory evaluation methods from the usability world. Bob Dick, from the Psychology department of Australia's University of Queensland, describes it thusly:

> Action research consists of a family of research methodologies which pursue action and research outcomes at the same time. It therefore has some components which resemble consultancy or change agency, and some which resemble field research.

See Bob's article, and the resources listed below, for more information.

Who can tell me more?

Click on any of the following links for more information:

Dick, Bob, A Beginner's Guide to Action Research, online web page at: http://ousd.k12.ca.us/netday/links/Action_Research/begin_guide_action_research. Great introduction to action research.

Gabel, Dorothy, An Introduction to Action Research, online web page at http://www.phy.nau.edu/~danmac/actionrsch.html. Another great primer on action research.

Giordano, Victoria, Evaluation in Instructional Systems Development, online web page at http://connect.barry.edu/ect607/SummEval.html. Victoria discusses instructional systems development models and their corresponding evaluation methodologies, including the Stake, Stufflebeam, and Scriven models. Her bibliography cites some of the works listed below.

Stake, R. E. (1967). The countenance of educational evaluation. In Ely, D. P. and Plomp, T. (Eds.) (1996). Classic Writings on Instructional Technology. Englewood, CO: Libraries Unlimited.

Stufflebeam, D. L. and Shinkfield, A. J. (1990). Systematic Evaluation. Boston, MA: Kluwer-Nijhoff.

Stufflebeam, D. L. (1991). The CIPP model for program evaluation. In G. F. Madaus, M. Scriven, & D. L. Stufflebeam (Eds.) Evaluation Models: Viewpoints on Educational and Human Services Evaluation (pp. 117–141). Boston, MA: Kluwer-Nijhoff.

University of Michigan Action Research Resources page, online web page at http://www3.extension.umn.edu/people/fhoefer/educdsgn/actresrc.htm. Great list of links to information on action research.

Suggested Reading for Usability Testing

This document provides a list of suggested readings for individuals who want to conduct usability tests. Readings are listed in the following categories:

Paper or site titles are linked to Web sites if available. Please contact the owners of the material directly for copyright information.
Book titles are linked to Amazon.com. Amazon provides the best descriptions of each book, particularly because they allow readers to add their own comments about any book in their collection. Get the scoop on each book from fellow readers. These book links are provided in association with Amazon.com. And yes, if you end up buying a book linked from this page, I get a cut. Support your friendly neighborhood grad student...

If you want to hold a usability test soon, read...

If you are on a tight deadline and want to hold a usability test for your project now, read books and articles listed in the "HowTo" Information section later in this document. If you have time to read only one book, read the Handbook of Usability Testing.

If you want to learn more about usability testing, read...

If you know you'll want to conduct usability tests in the future, but not necessarily right now, read books and articles listed in the Background Information section as well as in the "HowTo" Information section. If you're interested in a particular aspect of the field, see the works listed in the Additional Information section for more information.

Background Information

The following books and articles provide background information about the fields of usability engineering and human factors in general. They may contain little or no information directly related to usability testing. However, they provide a background that helps to place testing in context.

Bailey, Robert W. 1982. Human Performance Engineering: A Guide for System Designers. New York, Prentice Hall, Inc.

Classic textbook of human factors engineering.

Laurel, Brenda, ed., The Art of Human-Computer Interface Design, 1990, Addison-Wesley, Reading, MA, ISBN 0-201-51797-3.

Broad overview of HCI. A bit outdated, but an interesting look back at the good ol' days (when the Mac was new) of UI design.

Nielsen, Jakob, Usability Engineering, 1993, Academic Press/AP Professional, Cambridge, MA
ISBN 0-12-518406-9 (paper)

Basic introduction to the entire field of usability. The chapter on usability testing basically condenses the "how to" books listed in the next section into a single chapter. The book is intended as a textbook for computer user interface designers, but has very little technical detail compared to more procedural or advanced works. Contains a very extensive bibliography.

Norman, Donald, The Design of Everyday Things, 1988, Basic Books, New York, NY ISBN 0-385-26774-6 (paper) formerly published as The Psychology of Everyday Things (POET), ISBN 0-465-06709-3

Entertaining, layman's introduction to usability and user interfaces. Through the discussion of commonplace, everyday UIs such as auto dashboards, door handles, and alarm clocks, Norman teaches the cognitive psychology behind a user's interaction with an object.
Also by Donald Norman:

Turn Signals are the Facial Expressions of Automobiles (ISBN 0-xxx-xxxxx-x)

Things that Make Us Smart (ISBN 0-201-62695-0)

Preece, Jenny, ed., A Guide to Usability, 1993, Addison-Wesley, Reading MA, ISBN 020162768X.

Slim book on HCI and usability, probably the precursor to the usability chapters in Preece's Human-Computer Interaction.

Preece, Jenny, ed., Human-Computer Interaction, 1994, Addison-Wesley, Reading MA, ISBN 0201627698.

Large text on HCI. The chapters on usability evaluation are great; wish I had this book before I started this site.

"How To" Information

The following books and articles provide practical information; procedures, caveats, and techniques, for holding usability tests.

USINACTS Tutorial

Online Web page at http://atwww.hhi.de/USINACTS/tutorial/index.html. Detailed tutorial describing usability methodology. Great job!

Dumas, JS, and Redish, Janice, A Practical Guide to Usability Testing, 1993, Ablex, Norwood, NJ ISBN 0-89391-991-8 (paper)

Detailed guidelines for conducting usability tests, but oriented toward formal lab testing.

Lindgaard, G., Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems, 1994, Chapman and Hall, London, U.K. ISBN 0-412-46100-5

Basic usability testing textbook, geared toward computer system design.

Rubin, Jeffrey, Handbook of Usability Testing, 1994, John Wiley and Sons, New York, NY ISBN 0-471-59403-2 (paper)

Step-by-step guidelines for conducting usability tests. Probably the best all-around resource for doc usability testers.

Additional Information

The following books and articles provide in-depth information about a specific subset of the usability engineering field. Many are textbooks or papers written for graduate-level courses, and therefore may be boring or long-winded.

Usability Testing Experiences

Bell, Brigham, Rieman, John, and Lewis, Clayton. "Usability Testing of a Graphical Programming System: Things We Missed in a Programming Walkthrough." Communications of the ACM volume/number unknown (1991): 7-12

Describes a program walkthrough and compares it to a usability test on the same code development environment.

Chartier, Donald A. "Usability Labs: The Trojan Technology."

Online web page at http://www.ac.com/eagle/trojan.htm. Introduction to usability labs.

Cline, June A., Omanson, Richard C., and Marcotte, Donald A. "ThinkLink: An Evaluation of a Multimedia Interactive Learning Project."

Online web page at http://www.ameritech.com/corporate/testtown/library/articles/thinklink.html. Describes the a usability test of a multimedia project.

Haigh, Ruth, and Rogers, Andrew. "Usability Solutions for a Personal Alarm Device." Ergonomics In Design (July 1994): 12-21

Usability evaluation of a personal alarm design.

Heller, Hagan, and Ruberg, Alan. "Usability Studies on a Tight Budget." Design+Software: Newsletter of the ASD (1994)

Online Web page at http://www-pcd.stanford.edu/asd/info/articles/guerrilla-labs.html. Low-budget usability testing ideas.

Jordan, Patrick W., Thomas, Bruce, Weerdmeester, Bernard, (Eds.), Usability Evaluation in Industry, 1996, Taylor & Francis, Inc., London, UK. ISBN: 0-74-840460-0

Similar to the Wiklund book in that it describes industry practitioners' experiences, but more up to date.

Kirsner, Scott, "Listen Up-The Main Attraction," April 1, 1998, CIO Magazine.

Techniques used by leading Web sites to gather user information.

Lund, Arnold M. "Ameritech's Usability Laboratory: From Prototype to Final Design."

Online web page at http://www.ameritech.com/corporate/testtown/library/articles/uselab.html. Introduction to Ameritech's usability lab.

Whiteside, John, Bennett, John, and Holtzblatt, Karen. "Usability Engineering: Our Experience and Evolution" from Handbook of Human-Computer Interaction, M. Helander (ed.). Elsevier Science Publishers B.V. (North Holland), 1988: 791-804.

Describes the authors' practical experiences implementing usability engineering at IBM and DEC.

Wiklund, Michael E., Usability in Practice, 1994, AP Professional, Cambridge, MA ISBN 0-12-751250-0

Usability specialists from 17 companies describe how their usability programs got started, and how they incorporate usability engineering into the design of their products or services.

Yuschick, Matt, Schwab, Eileen, and Griffith, Laura. "ACNA--The Ameritech Customer Name and Address Service."

Online web page at http://www.ameritech.com/corporate/testtown/library/articles/acna.html. Describes the a user-centered design of a telephone company directory project.

Usability Testing Techniques and Issues

Bailey, R. W., et. al. "Usability Testing versus Heuristic Evaluation: A Head-to-Head Comparison." Proceedings of the Human Factors Society 36th Annual Meeting, (1992): 409-413.

Dayton, Tom, et. al. "Skills Needed By User-Centered Design Practitioners in Real Software Development Enironments: Report on the CHI `92 Workshop." SIGCHI Bulletin v25 n3, (July 1993): 16-31.

Describes the skills deemed most important for UCD practitioners--contains the fact that presentation skills rates very high, etc.

Jeffries, R., et. al., "User Interface Evaluation in the Real World: A Comparison of Four Techniques." Reaching through Technology: Proceedings of the 1991 CHI Conference, New Orleans, April-May 1991, NY: Association for Computing Machinery (ACM), 119-124.

Virzi, Robert A. "Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough?" Human Factors, v34, n4 (1992): 457-468.

Describes usability tests conducted with varying numbers of subjects, and their respective outcomes. Suggests using 22 subjects to identify any problem experienced by 10% or more of the population at the 90% confidence level, or 15 subjects if an 80% confidence level is acceptable.

Cost-Justifying Usability

Bias, R., and Mayhew, D. J., Cost-Justifying Usability, 1994, AP Professional, Cambridge, MA
ISBN 0-12-095810-4

Discusses the business case for usability work, including accounting for intangible benefits.

Gross, Clifford M., The Right Fit : The Power of Ergonomics As a Competitive Strategy, 1996, Productivity Press, ISBN: 156-327111-7.

Provides a Total Quality Management (TQM)-esqe look at ergonomics as product differentiator.

Karat, Claire-Marie. "Cost-Justifying Human Factors Support on Software Development Projects." Human Factors Society Bulletin v35, n11 (November 1992): 1-4.

Basic overview of the cost-benefit of including HF design considerations in the software development process.

Kirchner-Dean, Erica, and Kahler, Susan. "Marketing Usability: A Golden Opportunity." Ergonomics In Design (October 1994): 21-24

Usability concerns for software vendors; trade magazine's usability labs.

Mantei, Marilyn M., and Teorey, Toby J. "Cost/Benefit Analysis for Incorporating Human Factors in the Software Lifecycle." Communications of the ACM v31, n4 (April 1988): 428-439.

Overview of the cost-benefit of including HF design considerations in the software development process, including extensive balance sheets and accounting worksheets.

Nielsen, Jakob, "Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier"

Online Web page at http://www.useit.com/papers/guerrilla_hci.html. Was one of the chapters in Bias and Mayhew's Cost-Justifying Usability.

Low-Fidelity Prototyping

Cooper, Alan, "The Perils of Prototyping," August, 1994, Visual Basic Programmers Journal.

When the father of Visual Basic says that you shouldn't use VB for prototyping, well, you listen. Online Web page at http://www.cooper.com/articles/vbpj_perils_of_prototyping.html.

Nielsen, Jakob, "Paper versus Computer Implementations as Mockup Scenarios for Heuristic Evaluation", Human-Computer Interaction-Interact `90, D. Diaper et. al. (ed.) Elsevier Science Publishers B.V. (North Holland), 1990: 315-320

Nielsen provides a taxonomy of UI scenarios including different forms of mockups, then describes a heuristic evaluation usability test where the same videotex system was implemented as a paper mockup and as a HyperCard running prototype.

Rettig, Marc, "Prototyping for Tiny Fingers (Everything I Need to Know About Prototyping, I Learned In Kindergarten)", Communications of the ACM, April 1994.

Also available online at http://www.ac.com/eagle/lofidoc.htm. Humorous but concise precis of low-fidelity prototyping. [Note: Link broken as of 4/28/96]

Snyder, Carolyn, "Using Paper Prototypes to Manage Risk," October 1996, Software Design and Publisher Magazine

Online Web page at http://world.std.com/~uieweb/paper.htm. Many more good articles at UIE too.

Virzi, Robert A, Sokolov, Jeff, and Karis, Demetrios. "Usability Problem Identification Using Both Low- and High-Fidelity Prototypes," 1995: Obtained directly from the authors.

Study of two products using both high and low-fidelity prototypes. Study showed that low-fidelity prototypes can be effective throughout the product development cycle, not just during the initial stages of design.

User-Centered Design

Borenstein, Nathaniel S., Programming As If People Mattered : Friendly Programs, Software Engineering, and Other Noble Delusions, 1991, Princeton University Press, Princeton, NJ. ISBN: 0-691-03763-9

Mythical Man-Month-like collection of essays, describing various HCI issues in SW design.

Burgess, John H., Designing for Humans : The Human Factor in Engineering, 1986, Petrocelli Books, Princeton, NJ. ISBN: 0-89433-278-3.

Human factors text, but skewed toward evaluation and measurement than guidelines.

Carroll, John, (ed), Scenario-Based Design : Envisioning Work and Technology in System Development, 1995, John Wiley & Sons; ISBN: 0471076597.

How to use user scenarios to guide design.

Catterall, Bernard J. "The HUFIT Functionality Matrix." Human-Computer Interaction-Interact `90, D. Diaper et. al. (ed.) Elsevier Science Publishers B.V. (North Holland), 1990: 377-381.

The matrix cross-references user requirements with the technical proposal. Seems like Quality-Function Deployment for usability.

Gould, John D., Boies, Stephen J., and Lewis, Clayton. "Making Usable, Useful, Productivity-Enhancing Computer Applications." Communications of the ACM v34, n1 (January, 1991).

Usability design/iterative design process.

Grudin, Jonathan. "Interactive Systems: Bridging the Gaps Between Developers and Users." IEEE Computer, (April 1991): 59-69.

Describes three development contexts for interactive software development projects: competitively bid, commercial product, and in-house/custom development.

Lund, Arnold M., and Tschirgi, J. E., "Designing for People: Integrating Human Factors Into the Product Realization Process."

Online web page at http://www.ameritech.com/corporate/testtown/library/articles/design.html. Introduction to user-centered design.

Taylor, B. "HUFIT User Requirements Toolset." Proceedings of the Ergonomics Society Annual Conference: Contemporary Ergonomics 1989: 82-86.

Describes a set of tools to help develop user requirements and guidance on gathering and using user data.

"User Centered Design." Slides from User Centered Design Conference, Toronto, Canada, 20-24 February 1995.

Presentation from IBM Usability group giving an overview of user-centered design.

Contextual Inquiry

Beyer, Hugh, "Getting Started with Contextual Techniques".

Online Web page at http://www.incent.com/connection.indx/techniques.html. Outline-style notes (talking points from a presentation?) on contextual techniques.

Beyer, Hugh, and Holtzblatt, Karen, "Apprenticing with the Customer: A Collaborative Approach to Requirements Definition," Communications of the ACM, May 1995.

Online Web page at http://www.incent.com/papers.indx/requirements.html. One of the first writeups on Contextual Inquiry.

Beyer, Hugh, and Holtzblatt, Karen, Contextual Design : A Customer-Centered Approach to Systems Designs, 1997, Morgan Kaufman Publishers, ISBN: 1558604111

Finally, a book on contextual inquiry! From the inventors of the method, too. Don Norman sent in a review of the book to Amazon.

Contextual Connection email list: to subscribe to the Contextual Connection send mail to contextual-connection-request@world.std.com with "subscribe" in the body of the message.

From Beyer's description on his Web page: "Contextual-Connection is a moderated digest devoted to the discussion of design from customer data. It covers process, team, and organizational issues in gathering data about customers in the context of their work and the use of that data in to drive design."

Holtzblatt, Karen, and Beyer, Hugh, "Making Customer-Centered Design Work for Teams," Communications of the ACM, October 1993.

Online Web page at http://www.incent.com/papers.indx/Customer_Des_Teams.html. First published article on Contextual Design.

Holtzblatt, K., and Jones, S. "Contextual Inquiry: A Participatory Technique for System Design." in Schuler, D., and Namioka, A. (eds.) Participatory Design: Principles and Practice. Lawrence Earlbaum, Hillsdale, NJ. 1993: 180-193.

Describes methods, and principles for contextual inquiry.

Field Studies

Hackos, Jo Ann, and Redish, Ginny, 1998, User and Task Analysis for Interface Design, John Wiley & Sons; ISBN: 0471178314.

Huge, all-encompassing book on user and task analysis. Lots of information about site visits.

Kane, Kate, "Anthropologists Go Native In The Corporate Village," October/November 1996, Fast Company magazine.

Nouveau-biz zine article on ethnographers finding work in corporations.

Macht, Joshua, "The New Market Research," July 1998, Inc. magazine.

Biz-zine article on using low-tech field studies instead of expensive focus groups for market and user research.

Tamakoshi, Laura, 1996, Field Methods, and Field Notes.

Online Web pages at http://www.truman.edu/academics/ss/faculty/tamakoshil/. Describes Tamakoshi's methods used during her anthropological research in Papua New Guinea. Geared toward anthropology folks, but still useful.

Wixon, D. , and Ramey, Judith (Eds.), 1996, Field Methods Casebook for Software Design, John Wiley & Sons, Inc., New York, NY. ISBN: 0-471-14967-5

Nice overview of ethnographic techniques used in software design--the only book on these techniques as used for software design that I've seen so far. Similar to the Wiklund book in that it discusses individual practitioners' experiences.

Interviews and Focus Groups

Greenbaum, Thomas L.,"Focus Groups: A Help or a Waste of Time?" from Tom's articles page at http://www.groupsplus.com/.

Lots of other good focus group articles on that site. Greenbaum also has a couple of articles on video focus groups: "Focus Group By Video - Next Trend Of The 90's" and "Is Your Moderator Ready for Videoconferencing?" on his site.

Greenbaum, Thomas L., The Handbook for Focus Group Research, 1997, Sage Pubns; ISBN: 0761912533

Nielsen, Jakob, "The Use and Misuse of Focus Groups"

Templeton, Jane F., The Focus Group : A Strategic Guide to Organizing, Conducting and Analyzing the Focus Group Interview, 1994, Probus Pub Co; ISBN: 1557385300

Maya Design and W3Focus do online focus groups: FAQ

Maya Design also does telephone focus groups.

Silverman, George, "How to Get Beneath the Surface in Focus Groups" from George's articles page at http://www.mnav.com/index.html.

Lots of great focus group articles here, too.

Remote Evaluation

Hartson, Rex, Remote Evaluation Methods, 2002

Professor Hartson at Virginia Tech is taking over the content José Castillo used to have on his page, cited below. (http://research.cs.vt.edu/usability/projects/remote%20evaln/remote%20evaln.htm)

Castillo, José, Remote Usability Evaluation Home Page, 1998.

José has a ton of remote evaluation stuff on his page (http://hci.ise.vt.edu/~josec/remote_eval/index.html).

Surveys and Questionaires

Alreck, Pamela L., and Settle, Robert B., The Survey Research Handbook, 1994, Irwin Professional Publishing, Chicago, IL, ISBN: 0-256-10321-6 (ISBN is for cheaper paperbound 1995 version)

Nice general introduction to survey research, aimed at lower-division marketing and social sciences students. Less theoretical and more guideline-oriented than other texts.

Foddy, William, Constructing Questions for Interviews and Questionnaires: Theory and Practice in Social Research, Cambridge Univ Pr (Pap Txt); ISBN: 0521467330.

Gaddis, Susanne E., "How to Design Online Surveys," June 1998, Training and Development magazine, vol. 52, no. 6.

Lessler, Judith L., Questionnaire Design in the Cognitive Research Laboratory, ISBN 0840604122.

Oppenheim, A. N., Questionnaire Design, Interviewing and Attitude Measurement, 1992, Pinter Pub Ltd; ISBN: 1855670445

Salant, Priscilla, and Dillman, Don A., How to Conduct Your Own Survey, 1994, John Wiley & Sons, New York, NY, ISBN: 0471012734

Even less theoretical than the Alreck book.

Cognitive Walkthroughs

Rowley, David E., and Rhoades, David G. "The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure." CHI `92 Proceedings, (May 3-7, 1992): 389-395.

Describes a user interface evaluation method faster than regular unstructured walkthrough procedures.

Wharton, Cathleen, et. al., "The Cognitive Walkthrough Method: A Practictioner's Guide." in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Nice how-to article for cognitive walkthroughs.

Heuristic Evaluation

Instone, Keith, "Site Usability Evaluation".

Online Web page at http://webreview.com/wr/pub/97/10/10/usability/index.html. Instone describes how to do heuristic evaluation on a Web site.

Instone, Keith, "Usability Heuristics for the Web".

Online Web page at http://webreview.com/wr/pub/97/10/10/usability/sidebar.html. Instone revisits Nielsen's list of heuristics and adds Web-specific comments.

Nielsen, Jakob, "Heuristic Evaluation: How-To".

Online Web page at http://www.useit.com/papers/heuristic/heuristic_evaluation.html. How to do heuristic evaluation from the main proponent himself.

Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

Basic introduction to usability inspections, including heuristic evaluation, cognitive walkthroughs, and pluralistic walkthroughs.

Ergonomics

Bridger, R. S., Introduction to Ergonomics, 1995, McGraw Hill, New York, NY ISBN 0-07-007741-x

Introductory text on ergonomics, focusing on more on human factors than on workplace ergonomics. Good introductory chapters on human-machine interaction.

Pulat, B. Mustafa, Fundamentals of Industrial Ergonomics, 1992 (update July 1997), Prentice-Hall, Inc., Englewood Cliffs, NJ. ISBN 0-13-345364-2.

Pulat, B. Mustafa, and Alexander, David C., ed., Industrial Ergonomics: Case Studies, 1991, Industrial Engineering and Management Press, Norcross, GA, ISBN 0-89806-099-0.

Two texts on industrial ergonomics. Both are slim on evaluation techniques, although the case studies book does provide examples of workplace evaluations.

Motion Analysis

Barnes, Ralph M., Motion and Time Study: Design and Measurement of Work, 1980, John Wiley and Sons, New York, NY. ISBN 0471059056

Niebel, Benjamin, Motion and Time Study, 1984 (update Aug 1998), Richard D. Irwin, Inc., Homewood, IL. ISBN 02560257274

Two classic texts on motion analysis and time study. Predictive evaluation techniques that plot the user's interaction with an interface have their roots in motion analysis, a field popularized by Frank and Lillian Gilbreth (of Cheaper By the Dozen fame).

Grillo, Elmer V., and Berg, Charles J., Work Measurement in the Office, 1959, McGraw Hill, New York, NY. ISBN 5813869.

Ancient text covering motion analysis and time study, but with a focus on office tasks. Different tack than today's analogous texts that focus on injury-related issues of office work.

Statistics and Data Design

Box, George E. P., Hunter, William G., and Hunter, J. Stuart. 1978. Statistics for Experimenters. New York: John Wiley and Sons, Inc.

Textbook for statistical test design.

Corbetta, Piergiorgio 2003. Social Research—Theory, Methods and Techniques. SAGE Publ., ISBN 0761972536

Walpole, Ronald E., and Myers, Raymond H. 1985. Probability and Statistics for Engineers and Scientists. New York, Macmillan Publishing Co.

Textbook for statistical analysis of experimental data.

Young, Hugh D. 1962. Statistical Treatment of Experimental Data. New York: McGrawHill Co., Inc.

Trade paperback overview of statistical analysis of experimental data.

Summarizing Test Results

Brassard, Michael, ed. 1988. The Memory Jogger: A Pocket Guide of Tools for Continuous Improvement. Methuen, MA: Goal/QPC.

Also by Brassard:

The Memory Jogger II : A Pocket Guide of Tools for Continuous Improvement & Effective Planning, 1994

Quick-reference guide to charting and graphing tools used to summarize test results, including Pareto, pie, run, and x-y charts, histograms, means and standard deviations.

Presenting Test Results

The Dayton, et. al. study from CHI '92 found that presentation skills were perhaps the most important for usability practitioners. Remember, the goal of evaluation is to effect changes in the product--you can find all the problems you want, but if they're not fixed, your user isn't going to benefit. Here's a few sources for more information on presenting your results:

Information Graphics

Tufte, Edward R. 1990. Envisioning Information. Cheshire, CT: Graphics Press.

Tufte, Edward R. 1992. The Visual Display of Quantitative Information. Cheshire, CT: Graphics Press.

Tufte, Edward R. 1997. Visual Explanations : Images and Quantities, Evidence and Narrative. Cheshire, CT: Graphics Press.

All of Tufte's books show great examples of how graphic design can be used to describe quantitative information, and provide lots of ideas for incorporation into reports. Emphasis on convincing people (juries, Congress, etc.) using visual explanations.

Huff, Darrell. 1993. How to Lie with Statistics . New York, NY: W.W. Norton & Company, ISBN: 0393310728

A classic on how to skew opinions using the right graphs. Not like you'd need to do that, though, right?

Meyer, Eric K. 1997. Designing Infographics. New York, NY: Hayden Books, ISBN: 1568303394

Spence, Robert 2001. Information Visualization. Addison-Wesley, ISBN 0201596261

Presentations

Presentations Magazine
Online Web page at http://www.presentations.com/. Geared toward selling you a fancy multimedia projector, but also has good articles on delivery or presentation slide design.

Toastmasters International

Online Web page at http://www.toastmasters.org/. Nope, nothing to do with bread or appliances. One of the best resources for getting your speaking and presenting skills up to par. Join your local club today!

Kroeger, Lin. 1997. The Complete Idiot's Guide to Successful Business Presentations. New York, NY: Alpha Books, ISBN: 00-286-17-487

Quick and easy guide to presentations.

Kushner, Malcom. 1996. Successful Presentations for Dummies. IDG Books, ISBN: 1568843925

Same thing, this time from the Dummies folks.

Last Modified: 3 August 1998