CHAPTER

9

Usability testing on 10 cents a day

KEEPING TESTING SIMPLE—SO YOU DO ENOUGH OF IT

Why didn't we do this sooner?

—WHAT EVERYONE SAYS AT SOME POINT DURING THE FIRST USABILITY TEST OF THEIR WEB SITE

About once a month, I get one of these phone calls:

As soon as I hear "launching in two weeks" (or even "two months") and "usability testing" in the same sentence, I start to get that old fireman-headed-into-the-burning-chemical-factory feeling, because I have a pretty good idea of what's going on.

If it's two weeks, then it's almost certainly a request for a disaster check. The launch is fast approaching and everyone's getting nervous, and someone finally says, "Maybe we better do some usability testing."

If it's two months, then odds are that what they want is to settle some ongoing internal debates—usually about something very specific like color schemes. Opinion around the office is split between two different designs; some people like the sexy one, some like the elegant one. Finally someone with enough clout to authorize the expense gets tired of the arguing and says, "All right, let's get some testing done to settle this."

And while usability testing will sometimes settle these arguments, the main thing it usually ends up doing is revealing that the things they were arguing about aren't all that important. People often test to decide which color drapes are best, only to learn that they forgot to put windows in the room. For instance, they might discover that it doesn't make much difference whether you go with the horizontal navigation bar or the vertical menus if nobody understands the value proposition of your site.

Sadly, this is how most usability testing gets done: too little, too late, and for all the wrong reasons.

Repeat after me: Focus groups are not usability tests.

Sometimes that initial phone call is even scarier:

When the last-minute request is for a focus group, it's usually a sign that the request originated in Marketing. When Web sites are being designed, the folks in Marketing often feel like they don't have much clout. Even though they're the ones who spend the most time trying to figure out who the site's audience is and what they want, the designers and developers are the ones with most of the hands-on control over how the site actually gets put together.

As the launch date approaches, the Marketing people may feel that their only hope of sanity prevailing is to appeal to a higher authority: research. And the kind of research they know is focus groups.

I often have to work very hard to make clients understand that what they need is usability testing, not focus groups. Here's the difference in a nutshell:

- In a **focus group**, a small group of people (usually 5 to 8) sit around a table and react to ideas and designs that are shown to them. It's a group process, and much of its value comes from participants reacting to each other's opinions. Focus groups are good for quickly getting a sampling of user's opinions and feelings about things.
- In a **usability test**, one user at a time is shown something (whether it's a Web site, a prototype of a site, or some sketches of individual pages) and asked to either (a) figure out what it is, or (b) try to use it to do a typical task.

Focus groups can be great for determining what your audience wants, needs, and likes—in the abstract. They're good for testing whether the idea behind the site makes sense and your value proposition is attractive. And they can be a good way to test the names you're using for features of your site, and to find out how people feel about your competitors.

But they're *not* good for learning about whether your site works and how to improve it.

The kinds of things you can learn from focus groups are the things you need to learn early on, *before* you begin designing the site. Focus groups are for EARLY in the process. You can even run them late in the process if you want to do a reality check and fine-tune your message, but *don't* mistake them for usability testing. They *won't* tell you whether people can actually use your site.

Several true things about testing

Here are the main things I know about testing:

- **If you want a great site, you've got to test**. After you've worked on a site for even a few weeks, you can't see it freshly anymore. You know too much. The only way to find out if it really works is to test it.

  Testing reminds you that not everyone thinks the way you do, knows what you know, uses the Web the way you do.

I used to say that the best way to think about testing was that it was like travel: a broadening experience. It reminds you how different—and the same—people are, and gives you a fresh perspective on things.

But I finally realized that testing is really more like having friends visiting from out of town. Inevitably, as you make the tourist rounds with them, you see things about your home town that you usually don't notice because you're so used to them. And at the same time, you realize that a lot of things that you take for granted aren't obvious to everybody.

- **Testing one user is 100 percent better than testing none.** Testing always works. Even the worst test with the wrong user will show you things you can do that will improve your site.
- **Testing one user early in the project is better than testing 50 near the end.** Most people assume that testing needs to be a big deal. But if you make it into a big deal, you won't do it early enough or often enough to get the most out of it. A simple test early—while you still have time to use what you learn from it—is almost always more valuable than a sophisticated test later.

  Part of the conventional wisdom about Web development is that it's very easy to go in and make changes. The truth is, it turns out that it's not that easy to make changes to a site once it's in use. Some percentage of users will resist almost any kind of change, and even apparently simple changes often turn out to have far-reaching effects, so anything you can keep from building wrong in the first place is gravy.
- **The importance of recruiting representative users is overrated.** It's good to do your testing with people who are like the people who will use your site, but it's much more important to test early and often. My motto—as you'll see—is "Recruit loosely, and grade on a curve."
- **The point of testing is not to prove or disprove something. It's to inform your judgment.** People like to think, for instance, that they can use testing to prove whether navigation system "a" is better than navigation system "b", but you can't. No one has the resources to set up the kind of controlled experiment you'd need. What testing *can* do is provide you with invaluable input which, taken together with your experience, professional judgment, and common sense, will make it easier for you to choose wisely—and with greater confidence—between "a" and "b."

- **Testing is an iterative process.** Testing isn't something you do once. You make something, test it, fix it, and test it again.
- **Nothing beats a live audience reaction.** One reason why the Marx Brothers' movies are so wonderful is that before they started filming they would go on tour on the vaudeville circuit and perform scenes from the movie, doing five shows a day, improvising constantly and noting which lines got the best laughs. Even after they'd settled on a line, Groucho would insist on trying slight variations to see if it could be improved.

Mrs. Teasdale (Margaret Dumont) and Rufus T. Firefly eavesdrop in *Duck Soup*.

Lost our lease, going-out-of-business-sale usability testing

Usability testing has been around for a long time, and the basic idea is pretty simple: If you want to know whether your software or your Web site or your VCR remote control is easy enough to use, watch some people while they try to use it and note where they run into trouble. Then fix it, and test it again.

In the beginning, though, usability testing was a very expensive proposition. You had to have a usability lab with an observation room behind a one-way mirror, and at least two video cameras so you could record the users' reactions *and* the thing they were using. You had to recruit a lot of people so you could get results that were statistically significant. It was Science. It cost $20,000 to $50,000 a shot. It didn't happen very often.

But in 1989 Jakob Nielsen wrote a paper titled "Usability Engineering at a Discount"[1] and pointed out that it didn't have to be that way. You didn't need a usability lab, and you could achieve the same results with a lot fewer users.

[1] *Proceedings of the Third International Conference on Human-Computer Interaction, Boston, MA, Sept. 1989. Apple's interface "evangelist" Bruce Tognazzini also deserves credit for spreading the word in his widely read article for Apple developers, "User Testing on the Cheap" (reprinted in* Tog on Interface*, Addison-Wesley, 1992).*

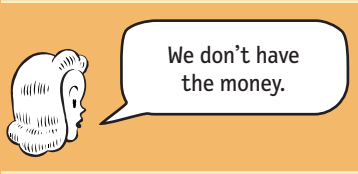
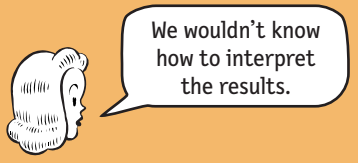
The idea of discount usability testing was a huge step forward. The only problem is that a decade later most people still perceive testing as a big deal, hiring someone to conduct a test still costs $5,000 to $15,000, and as a result it doesn't happen nearly often enough.

What I'm going to commend to you in this chapter is something even more drastic: Lost our lease, going-out-of-business-sale usability testing.

I'm going to try to explain how to do your own testing when you have *no* money and *no* time. If you can afford to hire a professional to do your testing, by all means do it—but *don't* do it if it means you'll do less testing.

	TRADITIONAL TESTING	LOST-OUR-LEASE TESTING
NUMBER OF USERS PER TEST	Usually eight or more to justify the set-up costs	Three or four
RECRUITING EFFORT	Select carefully to match target audience	Grab some people. Almost anybody who uses the Web will do.
WHERE TO TEST	A usability lab, with an observation room and a one-way mirror	Any office or conference room
WHO DOES THE TESTING	An experienced usability professional	Any reasonably patient human being
ADVANCE PLANNING	Tests have to be scheduled weeks in advance to reserve a usability lab and allow time for recruiting	Tests can be done almost any time, with little advance scheduling
PREPARATION	Draft, discuss, and revise a test protocol	Decide what you're going to show
WHAT/WHEN DO YOU TEST?	Unless you have a huge budget, put all your eggs in one basket and test once when the site is nearly complete	Run small tests continually throughout the development process
COST	$5,000 to $15,000 (or more)	About $300 (a $50 to $100 stipend for each user and $20 for three hours of videotape)
WHAT HAPPENS AFTERWARDS	A 20-page written report appears a week later, then the development team meets to decide what changes to make	Each observer writes one page of notes the day of the test. The development team can debrief the same day

THE TOP FIVE PLAUSIBLE EXCUSES FOR NOT TESTING WEB SITES

	It's true that most Web development schedules seem to be based on the punchline from a Dilbert cartoon. If testing is going to add to everybody's to-do list, if you have to adjust development schedules around tests and involve key people in preparing for them, then it won't get done. That's why you have to make testing as small a deal as possible. Done right, it will save time, because you won't have to (a) argue endlessly, and (b) redo things at the end.
	Forget $5,000 to 15,000. If you can convince someone to bring in a camcorder from home, you'll only need to spend about $300 for each round of tests.
	The least-known fact about usability testing is that it's incredibly easy to do. Yes, some people will be better at it than others, but I've never seen a usability test fail to produce useful results, no matter how poorly it was conducted.
	You don't need one. All you really need is a room with a desk, a computer, and two chairs where you won't be interrupted.
	It's true, the trickiest part of usability testing is making sure you draw the right conclusions from what you see. We'll cover that in the next chapter.

How many users should you test?

In most cases, I tend to think the ideal number of users for each round of testing is three, or at most four.

The first three users are very likely to encounter all of the most significant problems,[2] and it's much more important to do more rounds of testing than to wring everything you can out of each round. Testing only three users helps ensure that you *will* do another round soon.[3]

Also, since you will have fixed the problems you uncovered in the first round, in the next round it's likely that all three users will uncover a new set of problems, since they won't be getting stuck on the first set of problems.

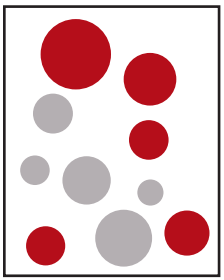
Testing only three or four users also makes it possible to test and debrief in the same day, so you can take advantage of what you've learned right away. Also, when you test more than four at a time, you usually end up with more notes than anyone has time to process—many of them about things that are really "nits," which can actually make it harder to see the forest for the trees. It's better to stay focused on the biggest problems, fix them, and then test again as soon as possible.

---

[2] *Jakob Nielsen and Tom Landauer have shown that testing five users will tend to uncover 85 percent of a site's usability problems, and that there is a serious case of diminishing returns for testing additional users. See Jakob Nielsen's March 2000 Alertbox column "Why You Only Need to Test with 5 Users" at* www.useit.com.

[3] *If you're hiring someone to do the testing for you* and *money is no object, you might as well test six or eight users since the additional cost per user will be comparatively low. But* only *if it won't mean you'll do fewer rounds of testing.*

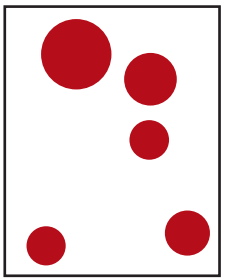
ONE TEST WITH 8 USERS



Eight users may find more problems in a single test.

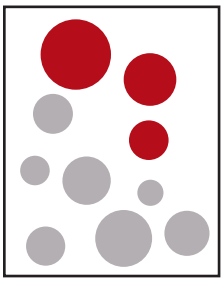
But the worst problems will usually keep them from getting far enough to encounter some others.

TOTAL PROBLEMS FOUND: 5



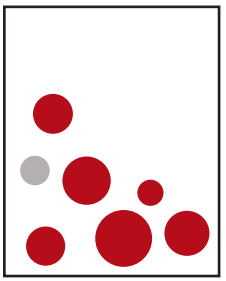
TWO TESTS WITH 3 USERS

First test



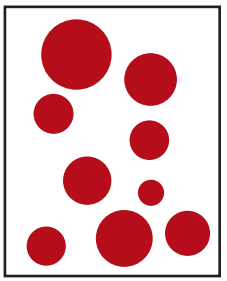
Three users may not find as many problems in a single test.

Second test



But in the second test, with the first set of problems fixed, they'll find problems they couldn't have seen in the first test.

TOTAL PROBLEMS FOUND: 9



Recruit loosely and grade on a curve

When people decide to test, they often spend a lot of time trying to recruit users who they think will precisely reflect their target audience—for instance, male accountants between the ages of 25 and 30 with one to three years of computer experience who have recently purchased expensive shoes.

The best-kept secret of usability testing is the extent to which *it doesn't much matter who you test*.

For most sites, all you really need are people who have used the Web enough to know the basics.

If you can afford to hire someone to recruit the participants for you *and* it *won't* reduce the number of rounds of testing that you do, then by all means be as specific as you

want. But if finding the ideal user means you're going to do fewer tests, I recommend a different approach:

Take anyone you can get (within limits) and grade on a curve.

In other words, try to find users who reflect your audience, but don't get hung up about it. Instead, try to make allowances for the differences between the people you test and your audience. I favor this approach for three reasons:

- **We're all beginners under the skin.** Scratch an expert and you'll often find someone who's muddling through—just at a higher level.
- **It's usually not a good idea to design a site so that only your target audience can use it.** If you design a site for accountants using terminology that you think all accountants will understand, what you'll probably discover is that a small but not insignificant number of accountants won't know what you're talking about. And in most cases, you need to be addressing novices as well as experts anyway, and if your grandmother can use it, an expert can.
- **Experts are rarely insulted by something that is clear enough for beginners.** Everybody appreciates clarity. (True clarity, that is, and not just something that's been "dumbed down.")

The exceptions:

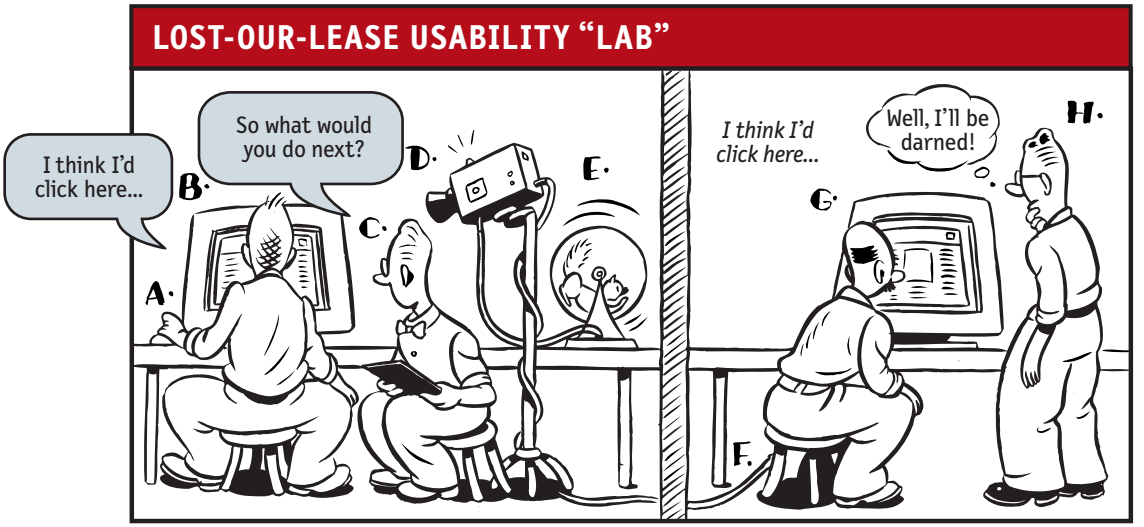
- **If your site is going to be used almost exclusively by one type of user and it's no harder to recruit from that group,** then do it. For instance, if your audience will be almost entirely women, then by all means test just women.
- **If your audience is split between clearly defined groups with very divergent interests and needs,** then you need to test users from each group at least once. For instance, if you're building a university site, for at least one round of testing you want to recruit two students, two professors, two high school seniors, and two administrators. But for the other rounds, you can choose any mix.
- **If using your site requires specific domain knowledge** (e.g., a currency exchange site for money management professionals), then you need to recruit people with that domain knowledge for at least one round of tests. But don't do it for every round if it will reduce the number of tests you do.

When you're recruiting:

- **Offer a reasonable incentive.** Typical stipends for a one-hour test session range from $50 for "average" Web users to several hundred dollars for professionals from a specific domain, like cardiologists for instance. I like to offer people a little more than the going rate, since (a) it makes it clear that I value their opinion, and (b) people tend to show up on time, eager to participate. Remember, even if the session is only 30 minutes, people usually have to block out another hour for travel time. Also, I'd rather have people who are curious about the process than people who are desperate for the money.
- **Keep the invitation simple.** "We need to have a few people look at our Web site and give us some feedback. It's very easy, and would take about forty-five minutes to an hour. And you'll be paid $___ for your time."
- **Avoid discussing the site (or the organization behind the site) beforehand.** You want their first look to tell you whether they can figure out what it is from a standing start. (Of course, if they're coming to your office, they'll have a pretty good idea whose site it is.)
- **Don't be embarrassed to ask friends and neighbors.** You don't have to feel like you're imposing if you ask friends or neighbors to participate. Most people enjoy the experience. It's fun to have someone take your opinion seriously and get paid for it, and they often learn something useful that they didn't know about the Web or computers in general.

Where do you test?

All you really need is an office or conference room with two chairs, a PC or Mac (with an Internet connection, if you're testing a live site), a camcorder, and a tripod.



Test subject (A) sits in front of computer monitor (B), while facilitator (C) tells him what to do and asks questions. Camcorder (D) powered by squirrel (E) is pointed at the monitor to record what the subject sees.

Meanwhile, cable (F) carries signal from camcorder to TV (G) in a nearby room where interested team members (H) can observe.

I recommend running a long cable from the camcorder to a TV in another office—or even a cubicle—nearby and encouraging everyone on the development team to come and watch.

The camcorder needs to record what the user sees (the computer screen or the designs on paper, depending on what you're testing) and what the user and the facilitator say. In most cases, you'll never go back and look at the videotapes, but they're good to have anyway, particularly to show to team members who want to observe but can't.

You can buy the camcorder, TV, cable, and tripod for less than $600. But if your budget won't stretch that far, you can probably twist somebody's arm to bring in a camcorder from home on test days.

Who should do the testing?

Almost anyone can facilitate a usability test; all it really takes is the courage to try it. With a little practice, most people can get quite good at it.

Try to choose someone who tends to be patient, calm, empathetic, a good listener, and inherently fair. Don't choose someone whom you would describe as "definitely not a people person" or "the office crank."

Who should observe?

Anybody who wants to. It's a good idea to encourage everyone—team members, people from marketing and business development, and any other stakeholders—to attend. If you can, try to get senior management to at least drop by; they'll often become fascinated and stay longer than they planned.

What do you test, and when do you test it?

The table on the next page shows the different kinds of testing you should do at each phase of Web development.

Before you even begin designing your site, you should be testing comparable sites. They may be actual competitors, or they may be sites that are similar in style, organization, or features to what you have in mind.

Use them yourself, then watch one or two other people use them and see what works and what doesn't. Many people overlook this step, but it's invaluable—like having someone build a working prototype for you for free.

If you've never conducted a test before testing comparable sites, it will give you a pressure-free chance to get the hang of it. It will also give you a chance to develop a thick skin. The first few times you test your own site, it's hard not to take it personally when people don't get it. Testing someone else's site first will help you see how people react to sites and give you a chance to get used to it.

Since the comparable sites are "live," you can do two kinds of testing: "Get it" testing and key tasks.

	PLANNING	ROUGH SKETCHES	PAGE DESIGNS	PROTOTYPE	FIRST USABLE VERSION	"CUBICLE TESTS"
WHAT TO TEST	Competitors' sites	Sketch of Home page Names of top level categories and site features	Home page Second-level page template Content page template	As much as you have working	As much as you have working	Each unique page
FORMAT	Live site	Paper	Paper	HTML prototype	Live site	HTML page
HOW TO TEST	"Get it" Key tasks	"Get it" Names of things	"Get it" Basic navigation	"Get it" Key tasks	"Get it" Key tasks	Key tasks
WHAT YOU'RE LOOKING FOR	What do they like/love? How does it fit into their lives? What works well? How hard is it to do key tasks?	Do they get the point of the site? Does it seem like what they need?	Do they get the point of the site? Do they get the navigation? Can they guess where to find things?	Do they still get it? Can they accomplish the key tasks?	Do they still get it? Can they accomplish the key tasks?	Can they accomplish the key tasks?
SESSION LENGTH	1 hr.	15-20 min.	15-20 min.	45 min.-1hr.	1 hr.	5 min. per page
# OF TESTS	1	1-3	1-3	1-3	1-3	1 per page

TOTAL BUDGET: 13 TESTS x 3 USERS PER TEST x $100 PER USER = $3900

- **"Get it" testing** is just what it sounds like: show them the site, and see if they get it—do they understand the purpose of the site, the value proposition, how it's organized, how it works, and so on.
- **Key task testing** means asking the user to do something, then watching how well they do.

  As a rule, you'll always get more revealing results if you can find a way to observe users doing tasks that they have a hand in choosing. It's much better, for instance, to say "Find a book you want to buy, or a book you bought recently" than "Find a cookbook for under $14." When people are doing made-up tasks, they have no emotional investment in it, and they can't use as much of their personal knowledge.

As you begin designing your own site, it's never too early to start showing your design ideas to users, beginning with your first rough sketches. Designers are often reluctant to show work in progress, but users may actually feel freer to comment on something that looks unfinished, since they know you haven't got as much invested in it and it's still subject to change. Also, since it's not a polished design, users won't be distracted by details of implementation and they can focus on the essence and the wording.

Later, as you begin building parts of the site or functioning prototypes, you can begin testing key tasks on your own site.

I also recommend doing what I call Cubicle tests: Whenever you build a new kind of page—particularly forms—you should print the page out and show it to the person in the next cubicle and see if they can make sense out of it. This kind of informal testing can be very efficient, and eliminate a lot of potential problems.

CHAPTER

10

Usability testing: The movie

HOW TO DO YOUR OWN TESTING

I don't like the colors.

—WHAT YOU CAN COUNT ON AT LEAST ONE USER SAYING IN EVERY USABILITY TEST

This chapter explains how to conduct a test if you're the facilitator and what to look for if you're an observer.

What to do if you're the facilitator

If you've never conducted a usability test, the main thing you need to know is that you should just relax, because there's not much to it. Your responsibility is to tell the users what you want them to do, to encourage them to think out loud, to listen carefully to what they have to say, and to protect them.

Here's a list of the things to keep in mind. If you read this list and the sample session that follows it, you'll be ready to start testing.[1]

- **Try the test yourself first.** The day before the test, try doing whatever you're going to ask the test participants to do and make sure that you can do it in the time allotted. Make sure that whatever pages you're testing are accessible from the computer you'll be using, and that you have any passwords you'll need.
- **Protect the participants.** It's your responsibility to prevent any damage to your test users' self-esteem. Be nice to them. If they get stuck, don't let them get too frustrated. Be sure to pat them on the back (figuratively), and thank them (sincerely) when you're done. Let them know that their participation has been very helpful—exactly what you needed.
- **Be empathetic.** Be kind, patient, and reassuring. Make it clear to them that you *know* they're not stupid.

[1] *If you still feel the need for more advice, I recommend Jeffrey Rubin's* Handbook of Usability Testing. *See my reading list on page 183.*

- **Try to see the thought balloons forming over their heads.** The main thing you're trying to do is observe their thought process. Whenever you're not sure what they're thinking, ask them. If the user has been staring at the screen for ten seconds, ask, "What are you looking at?" or "What are you thinking?"

  You're trying to see what their expectation is at each step and how close the site comes to matching that expectation. When they're ready to click, ask what they expect to see. After they click, ask if the result was what they expected.

- **Don't give them hints about what to do.** It's a lot like being a therapist. If they say, "I'm not sure what to do next," you should say, "What do *you* think you should do?" or "What would you do if you were at home?"

- **Keep your instructions simple.** There aren't very many, so you'll learn them quickly.

  > "Look around the page and tell me what you think everything is and what you would be likely to click on."
  >
  > "Tell me what you would click on next and what you expect you would see then."
  >
  > "Try to think out loud as much as possible."

  Don't be afraid to keep repeating them; it will be more boring to you than to the user.

- **Probe, probe, probe.** You have to walk a delicate line between distracting or influencing the users and finding out what they're really thinking, which they may not know themselves.

  For instance, when a user says, "I like this page" you always want to ask a leading question like "What do you like best about it?" If this produces "Well, I like the layout" then you need to follow with "What appeals to you about the layout?" You're looking for specifics, not because the specifics themselves are necessarily important but because eliciting them is the only way you can be sure you understand what the user is really reacting to.

- **Don't be afraid to improvise.** For instance, if the first two users get hopelessly stuck at the same point and it's obvious what the problem is and how to fix it, don't make the third user struggle with it needlessly. As soon as he encounters the problem, explain it and let him go on to something more productive.
- **Don't be disappointed if a user turns out to be inexperienced or completely befuddled.** You can often learn more by watching a user who *doesn't* get it than one who does. Because more experienced users have better coping strategies for "muddling through," you may not even notice that they don't get it.
- **Make some notes after each session.** Always take a few minutes right after each test session to jot down the main things that struck you. If you don't do it before you start the next test, it will be very hard to remember what they were.

A sample test session

Here's an annotated excerpt from a typical—but imaginary—test session. The site is real, but it has since been redesigned.[2] The participant's name is Janice, and she's about 25 years old.

[2] *My thanks to the folks at eLance.com for allowing me to use an earlier version of their site as the subject of this example.*

INTRODUCTION

Hi, Janice. My name is Steve Krug, and I'm going to be walking you through this session.

This whole first section is the script that I use when I conduct tests.[3]

You probably already know, but let me explain why we've asked you to come here today. We're testing a Web site that we're working on so we can see what it's like for actual people to use it.

I want to make it clear right away that we're testing the *site*, not you. You can't do anything wrong here. In fact, this is probably the one place today where you don't have to worry about making mistakes.

We want to hear exactly what you think, so please don't worry that you're going to hurt our feelings.[4] We want to improve it, so we need to know honestly what you think.

As we go along, I'm going to ask you to think out loud, to tell me what's going through your mind. This will help us.

I always have a copy in front of me, and I don't hesitate to read from it, but I find it's good to ad lib a little, even if it means making mistakes. When the users see that I'm comfortable making mistakes, it helps take the pressure off them.

[3] *A copy of the script is available on the companion Web site (www.stevekrug.com) so you can download it and edit it for your own use.*

[4] *If you didn't work on the part that's being tested, you can also say, "Don't worry about hurting my feelings. I didn't create the pages you're going to look at."*

If you have questions, just ask. I may not be able to answer them right away, since we're interested in how people do when they don't have someone sitting next to them, but I will try to answer any questions you still have when we're done. We have a lot to do, and I'm going to try to keep us moving, but we'll try to make sure that it's fun, too.	It's important to mention this, because it will seem rude not to answer their questions as you go along. You have to make it clear before you start that (a) it's nothing personal, and (b) you'll try to answer them at the end if they still want to know.
You may have noticed the camera. With your permission, we're going to videotape the computer screen and what you have to say. The video will be used only to help us figure out how to improve the site, and it won't be seen by anyone except the people working on the project. It also helps me, because I don't have to take as many notes. There are also some people watching the video in another room.	At this point, most people will say something like, "I'm not going to end up on *America's Funniest Home Videos*, am I?"
If you would, I'm going to ask you to sign something for us. It simply says that we have your permission to tape you, but that it will only be seen by the people working on the project. It also says that you won't talk to anybody about what we're showing you today, since it hasn't been made public yet. Do you have any questions before we begin? No. I don't think so.	Give them the release and non-disclosure agreement to sign. It should be as short as possible and written in plain English.[5]

[5] *You'll find a sample form on the companion Web site.*

BACKGROUND QUESTIONS

Before we look at the site, I'd like to ask you just a few quick questions. First, what's your occupation?

I'm a router.

I've never heard of that before. What does a router do, exactly?

Not much. I take orders as they come in, and send them to the right office.

Good. Now, roughly how many hours a week would you say you spend using the Internet, including email?

Oh, I don't know. Probably an hour a day at work, and maybe four hours a week at home. Mostly that's on the weekend. I'm too tired at night to bother. But I like playing games sometimes.

How do you spend that time? In a typical day, for instance, tell me what you do, at work and at home.

I find it's good to start with a few questions to get a feel for who they are and how they use the Internet. It gives them a chance to loosen up a little and gives you a chance to show that you're going to be listening attentively to what they say—and that there are no wrong or right answers.

Don't hesitate to admit your ignorance about anything. Your role here is not to come across as an expert, but as a good listener.

Notice that she's not sure how much time she really spends on the Internet. Most people aren't. Don't worry. Accurate answers aren't important here. The main point here is just to get her talking and thinking about how she uses the Internet and to give you a chance to gauge what kind of user she is.

Well, at the office I spend most of my time checking email. I get *a lot* of email, and a lot of it's junk but I have to go through it anyway. And sometimes I have to research something at work.

Do you have any favorite Web sites?

Yahoo, I guess. I like Yahoo, and I use it all the time. And something called Snakes.com, because I have a pet snake.

Really? What kind of snake?

A python. He's about four feet long, but he should get to be eight or nine when he's fully grown.

Wow. OK, now, finally, have you bought anything on the Internet? How do you feel about buying things on the Internet?

Don't be afraid to digress and find out more about the user, as long as you come back to the topic before long.

I've bought some things recently. I didn't do it for a long time, but only because I couldn't get things delivered. It was hard to get things delivered, because I'm not home during the day. But now one of my neighbors is home all the time, so I can.

And what have you bought?

Well, I ordered a raincoat from L.L. Bean, and it worked out *much* better than I thought it would. It was actually pretty easy.

OK, great. We're done with the questions, and we can start looking at things.

OK, I guess.

REACTIONS TO THE HOME PAGE

First, I'm just going to ask you to look at this page and tell me what you think it is, what strikes you about it, and what you think you would click on first.

For now, don't actually click on anything. Just tell me what you *would* click on.

And again, as much as possible, it will help us if you can try to think out loud so we know what you're thinking about.

The browser has been open, but minimized. At this point, I reach over and click to maximize it.

Well, I guess the first thing I notice is that I like the color. I like the shade of orange, and I like the little picture of the sun [at the top of the page, in the eLance logo].

Let's see. [Reads.] "The global services market." "Where the world comes to get your job done."

I don't know what that means. I have no idea.

"Animate your logo free." [Looking at the Cool Stuff section on the left.] "3D graphics marketplace." "eLance community." "eLance marketplace."

In an average test, it's just as likely that the next user will say that she hates this shade of orange and that the drawing is too simplistic. Don't get too excited by individual reactions to site aesthetics.

There's a lot going on here. But I have no idea what any of it is.

If you had to take a guess, what do you think it might be?

Well, it seems to have something to do with buying and selling...something.

[Looks around the page again.] Now that I look at the list down here [the Yahoo-style category list halfway down the page], I guess maybe it must be services. Legal, financial, creative...they all sound like services.

This user is doing a good job of thinking out loud on her own. If she wasn't, this is where I'd start asking her, "What are you thinking?"

So I guess that's what it is. Buying and selling services. Maybe like some kind of online *Yellow Pages*.

OK. Now, if you were at home, what would you click on first?

I guess I'd click on that 3D graphics thing. I'm interested in 3D graphics.

Before you click on it, I have one more question. What about these pictures near the top of the page—the ones with the numbers? What did you make of them?

I ask this question because the site's designers think most users are going to start by clicking on the pictures of the five steps, and that everyone will at least look at them.

I noticed them, but I really didn't try to figure them out. I guess I thought they were telling me what the steps in the process would be.

Any reason why you didn't pay much attention to them?

No. I guess I just wasn't ready to start the process yet. I didn't know if I *wanted* to use it yet. I just wanted to look around first.

OK. Great.

At this point, I would let her explore for a few minutes, following whatever links she's interested in and encouraging her to think out loud the whole time.

TESTING A TASK

OK, now we're going to try something else.

Can you think of something you might want to post as a project if you were using this site?

> Hmm. Let me think. I think I saw "Home Improvement" there somewhere. We're thinking of building a deck. Maybe I would post that.

So if you were going to post the deck as a project, what would you do first?

> I guess I'd click on one of the categories down here. I think I saw home improvement. [Looks.] There it is, under "Family and Household."

So what would you do?

> Well, I'd click.... [Hesitates, looking at the two links under "Family and Household."]

After a few minutes of "free range clicking," I give her a task to perform so we can see whether she can use the site for its intended purpose.

Whenever possible, I like to let the user have some say in choosing the task.

Well, now I'm not sure *what* to do. I can't click on Home Improvement, so it looks like I have to click on either "RFPs" or "Fixed-Price." But I don't know what the difference is.

Fixed price I sort of understand; they'll give me a quote, and then they have to stick to it. But I'm not sure what RFPs is.

As it turns out, she's mistaken. Fixed-price (in this case) means services available for a fixed hourly rate, while an RFP (or Request for Proposal) is actually the choice that will elicit quotes. This is the kind of misunderstanding that often surprises the people who built the site.

Well, which one do you think you'd click on?

Fixed price, I guess.

Why don't you go ahead and do it?

From here on, I just watch while she tries to post a project, letting her continue until either (a) she finishes the task, (b) she gets really frustrated, or (c) we're not learning anything new by watching her try to muddle through.

What to do if you're observing

Being an observer at a usability test is a very cushy job. All you have to do is listen, watch closely, keep an open mind, and take notes.

Here are the types of things you're looking for:

- **Do they get it?** Without any help, can the users figure out what the site or the page is, what it does, and where to start?
- **Can they find their way around?** Do they notice and understand the site's navigation? Does your hierarchy—and the names you're using for things—make sense to them?
- **Head slappers.** You'll know these when you see them: the user will do something, or *not* do something, and suddenly everyone who's observing the session will slap his or her forehead and say, "Why didn't we think of that?" or "Why didn't we ever notice that?" These are *very* valuable insights.
- **Shocks.** These will also make you slap your head, but instead of saying "Why didn't *we* notice that?" you'll say, "How could she [the user] *not* notice that?" or "How could she not understand that?" For instance, you might be shocked when someone doesn't notice that there is a menu bar at the top of each page, or doesn't recognize the name of one of your company's products.

  Unlike head slappers, the solution to shocks won't always be obvious and they may send you back to the drawing board.
- **Inspiration.** Users will often suggest a solution or the germ of a solution to a problem that you've struggled with for a long time. Very often the solution will be something you'd already thought about and rejected, but just watching someone actually encounter the problem will let you see it in a whole new light. And often something else about the project has changed in the meantime (you've decided to use a different technology, for instance, or there's been a shift in your business priorities) that makes an abandoned approach suddenly feasible.
- **Passion.** What are the elements of the site that users really connect with? Be careful not to mistake mere enthusiasm for passion, though. You're looking for phrases like "This is exactly what I've been looking for!" or "When can I start using this?"

In the course of any test, you'll also notice a number of things that are just not working like missing graphics, broken links, or typos. You should keep a list of these things so you can pass it on to whomever will fix them, but they're not what you're there to find, and you shouldn't let them distract you.

Here are some things to keep in mind when you're observing:

- **Brace yourself.** You may be disappointed by the users' reactions. Some people just won't get it. Some just won't like it. Some will get lost and confused, apparently without reason. It can be emotionally wrenching to watch someone have a negative reaction to something you've poured your soul into. The mantra you want to have in your head is not "It's not working!" but, rather, "What will it take to fix it?"
- **Don't panic.** Try to resist the temptation to jump to *any* conclusions until you've seen at least two users, preferably three.
- **Be quiet.** There's nothing more disconcerting for a test participant than the sound of laughter—or groans—coming from an adjoining room when she's having trouble using the site.
- **Remember that you're grading on a curve.** When a participant who uses the Internet two hours a day doesn't know how to type a URL, don't think, "Sheesh! What a dolt." Think, "How many people are there just like that out there? Can we afford to lose all of them as users?"
- **Remember that you're seeing their best behavior.** When you're watching a test you need to remember that people will tend to read Web pages much more thoroughly and put more effort into figuring things out in a test situation than they will in real life. After all, they're not under any time pressure, they're being paid to figure it out, and—most importantly—they don't want to look stupid. So when they can't figure something out, you have to realize that they're trying much harder than most people will and they *still* can't get it.
- **Pay more attention to actions and explanations than opinions.** Opinions expressed during user tests are notoriously unreliable. People will often exaggerate their opinions—positive *and* negative—because they think you *want* them to express strong opinions.

Reporting what you saw

As soon as possible after the test, each observer and the facilitator should type up a short list of the main problems they saw and any thoughts they have about how to fix them.

You don't want to write a comprehensive report, more like an executive summary. Ideally you want the entire development team to read all of these lists (or at least scan them), so each one should be no longer than a page or two.

Here's an example of the kind of notes I usually write:

USABILITY TEST NOTES: ELANCE.COM

Steve Krug

June 1, 2000

- Everybody seemed to be drawn immediately to the "Cool Stuff" links at the top of the Home page, particularly "Animate your logo: Free!" It's good that it engaged them, but after they were finished looking at the offer, they all seemed puzzled about how it related to the site.
- Two of the three users were unable to figure out what eLance was without some help. They figured it out eventually, but one of them said he wouldn't have bothered if he wasn't in a test. The wording of the main message still needs work.
- Everybody was unclear about how to get started. They all found a way, but they were uncertain and anxious along the way. There still may be too many entry points.
- Two users thought they understood the site, but they were puzzled by the Community tab. Is there another name we could use?
- The words "Buy" and "Sell" seemed to puzzle them. They weren't sure whether to think of themselves as buyers or sellers.
- Two of the three users were confused by RFP vs. Fixed-price.
- They all liked the category listing. Seeing categories they were interested in right away on the Home page encouraged them to go on and find out more.

CHAPTER

11

On not throwing the baby out with the dishes

INTERPRETING TEST RESULTS

Sure, we've made mistakes. But let's not throw the baby out with the dishes.

—LYNDON BAINES JOHNSON

OK, you've done your testing, and you've got everyone's notes. How do you decide what to change?

Review the results right away

After each round of tests, you should make time as soon as possible for the development team to review everyone's observations and decide what to do next.

At this meeting, you should distribute copies of everyone's notes and copies of whatever screens or sketches were tested. You're doing two things at this meeting:

- **Triage**—reviewing the problems people saw and deciding which ones need to be fixed.
- **Problem solving**—figuring out how to fix them.

It might seem that this would be a difficult process. After all, these are the same team members who've been arguing about the right way to do things all along. So what's going to make this session any different?

Just this:

> The important things that you learn from usability testing usually *just make sense*. They tend to be obvious to anyone who watches the sessions.

Also, the experience of seeing your handiwork through someone else's eyes will often suggest entirely new solutions for problems, or let you see an old idea in a new light.

And remember, this is a cyclic process, so the team doesn't have to agree on the perfect solution. You just need to figure out what to try next.

Can this marriage be saved?

Naturally, the biggest question on everyone's mind, especially during the first few tests, is always "Are we basically on the right track?"

At these debriefing meetings, you're always trying to figure out whether the part of the site that you're testing can be made to work by tweaking it, or if you have to scrap it and take a whole different approach. For instance, if some users don't get the whole idea of the site, does it mean that the basic concept is flawed, or do you just have to change some of the wording?

Here's my advice:

- **Always consider tweaking first.** It's rare that a site is completely off-base, but there's a tendency to be spooked by any bad user reactions, particularly after the first round of testing. Before scrapping anything, always stop and think, "What's the *least* we could do that might fix the problems we're seeing?" If it seems like a particular tweak has a reasonable chance of working, mock it up and test it as soon as possible.
- **Focus on specifics.** Try to avoid sweeping statements and focus on the precise points where people seemed to go astray. "They didn't seem to notice the navigation when they got to the second-level pages" is a much more productive place to begin problem solving than "The navigation didn't work." One suggests that you should think about ways you could make the navigation slightly more noticeable; the other suggests you should start considering a whole new navigation scheme.
- **Tweak, but verify.** Remember, this is a cyclic process. Since you'll be doing another test soon, you can afford to try some tweaks before scrapping anything.

On the other hand:

- **If the problem is deep, bite the bullet.** I sometimes walk into situations where the site has a serious fundamental problem that nobody is talking about. The problem has usually been obvious to everyone involved for a long time, but nobody wants to be the one to mention it because (a) it seems like there's no solution, and (b) it seems like it's too late to do anything but tough it out and hope for the best. Sometimes I get called into these situations precisely because they need an outsider to announce what's already obvious to everyone.

  For instance, suppose that when you do your first test it becomes clear that the way you've chosen to organize the site isn't the way your users would organize it, and as a result they're having a very hard time figuring out where you've put things. Or perhaps you've been building your company's site assuming that your customers will want to do *x* and *y*, but when you start testing you learn that they're happy with the way they already do *x* and *y* and they're not really interested in doing them online.

  If you've got a deep-seated problem, a post-test debriefing meeting is a good place to finally face it. Once it's out on the table, it usually turns out to be more manageable than it seems when you're not talking about it.

- **Remember: It's *almost* never too late to challenge basic assumptions.** Rethinking the basics doesn't necessarily mean changing everything, and it often turns out that the solution to the problem is simpler than you've feared.

Typical problems

Here are the types of problems you're going to see most often when you test:

- **Users are unclear on the concept.** They just don't get it. They look at the site or a page and they either don't know what to make of it, or they think they do but they're wrong.
- **The words they're looking for aren't there.** This usually means that either (a) the categories you've used to organize your content aren't the ones they would use, or (b) the categories are what they expect, but you're just not using the names they expect.
- **There's too much going on.** Sometimes what they're looking for is right there on the page, but they're just not seeing it. In this case, you need to either (a) reduce the overall noise on the page, or (b) turn up the volume on the things they need to see so they "pop" out of the visual hierarchy more.

Some triage guidelines

Here's the best advice I can give you about deciding what to fix—and what not to.

- **Ignore "kayak" problems.** In any test, you're likely to see several cases where users will go astray momentarily but manage to get back on track almost immediately without any help. It's kind of like rolling over in a kayak; as long as the kayak rights itself quickly enough, it's all part of the so-called fun. In basketball terms, no harm, no foul.

  As long as (a) everyone who has the problem notices that they're no longer headed in the right direction quickly, and (b) they manage to recover without help, and (c) it doesn't seem to faze them, you can ignore the problem. In general, if the user's second guess about where to find things is always right, that's good enough.

  Of course, if there's an easy and obvious fix that won't break anything else, then by all means fix it. But kayak problems usually don't come as a surprise to the development team. They're usually there because of some ambiguity for which there is no simple resolution. For example, there are usually at least one or two oddball items that don't fit perfectly into any of the top-level categories of a site. So half the users may look for movie listings in Lifestyles first, and the other half will look for them in Arts first. Whatever you do, half of them are going to be wrong on their first guess, but everyone will get it on their second guess, which is fine.[1]

- **Resist the impulse to add things.** When it's obvious in testing that users aren't getting something, most people's first reaction is to add something, like an explanation or some instructions.

  Very often, the right solution is to take something (or things) away that are obscuring the meaning, rather than adding yet another distraction.

[1] *You may be thinking "Well, why not just put it in both categories?" In general, I think it's best for things to "live" in only one place in a hierarchy, with a prominent "see also" crosslink in any other places where people are likely to look for them.*

- **Take "new feature" requests with a grain of salt.** People will often say, "I'd like it better if it could do *x*." It always pays to be suspicious of these requests for new features. If you probe deeper, it often turns out that they already have a perfectly fine source for *x* and wouldn't be likely to switch; they're just telling you what they like.
- **Grab the low-hanging fruit.** The main thing you're looking for in each round of testing is the big, cheap wins. These fall into two categories:
    - **Head slappers.** These are the surprises that show up during testing where the problem and the solution were obvious to everyone the moment they saw the first user try to muddle through. These are like found money, and you should fix them right away.
    - **Cheap hits.** Also try to implement any changes that (a) require almost no effort, or (b) require a *little* effort but are highly visible.

And finally, there's one last piece of advice about "making changes" that deserves its own section:

Don't throw the baby out with the dishes

Like any good design, successful Web pages are usually a delicate balance, and it's important to keep in mind that even a minor change can have a major impact. Sometimes the real challenge isn't fixing the problems you find—it's fixing them *without* breaking the parts that already work.

Whenever you're making a change, think carefully about what else is going to be affected. In particular, when you're making something more prominent than it was, consider what else might end up being de-emphasized as a result.

That's all, folks

As Bob and Ray used to say, "Hang by your thumbs, and write if you get work."

I hope you'll check in at the companion Web site www.stevekrug.com from time to time. I'm going to be working on some pseudo-research (as I like to think of it) using eye tracking equipment to learn more about what we actually see when we look at Web pages, and I'll try to post some observations now and then.

But above all, be of good cheer. As I said at the beginning, building a great Web site is an enormous challenge, and anyone who gets it even half right has my admiration.

And please don't take anything I've said as being against breaking "the rules"—or at least bending them. I know there are even sites where you want the interface to make people think, to puzzle or challenge them. Just be sure you know which rules you're bending, and that you at least *think* you have a good reason for bending them.