Usability testing on 10 cents a day

KEEPING TESTING SIMPLE — SO YOU DO ENOUGH OF IT

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

We don't have the time.	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.
We don't have the money.	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.
We don't have the expertise.	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.
We don't have a usability lab.	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.
We wouldn't know how to interpret the results.	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.

[145]

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.*
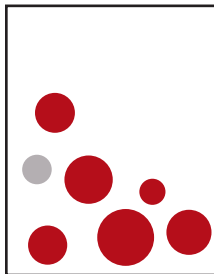
ONE TEST WITH 8 USERS
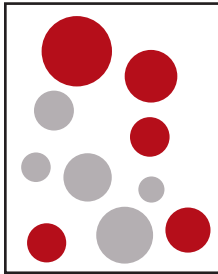
Eight users may find more problems in a single test.

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

TOTAL PROBLEMS FOUND: 5
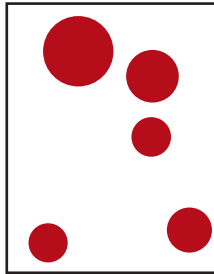
TWO TESTS WITH 3 USERS

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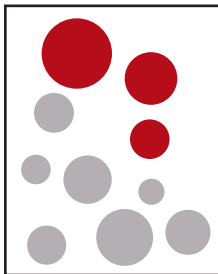
Three users may not find as many problems in a single test.

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
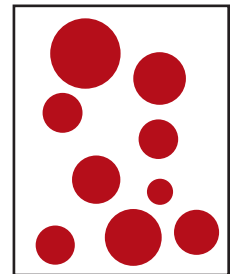
Recruit loosely and grade on a curve

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.

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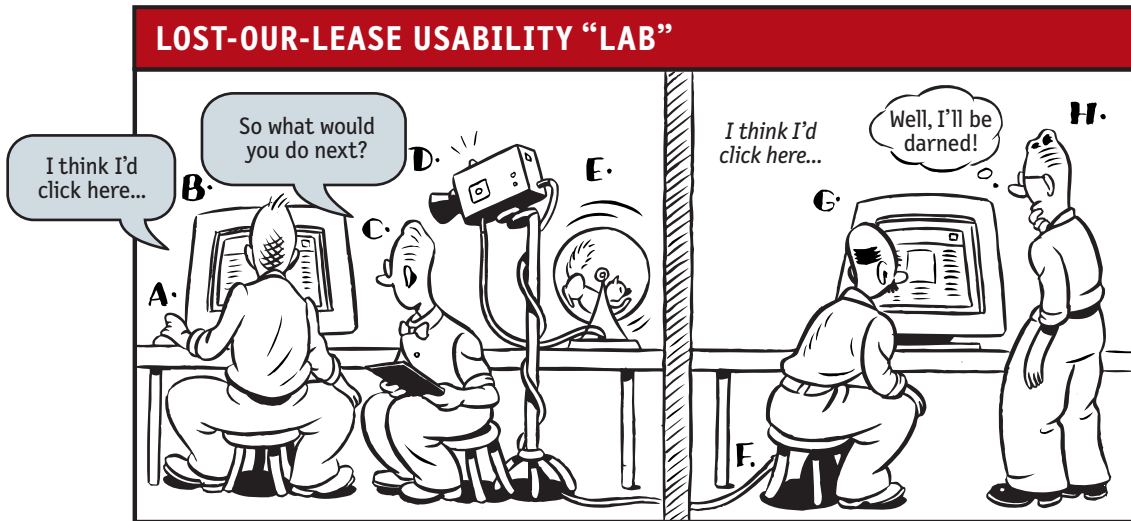
Test subject (A) sits in front of computer monitor (B), while facilitator (C) tells him what to do and asks questions. Camcorder (D) powered by squirrel (E) is pointed at the monitor to record what the subject sees.

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.