Chapter 4. What do you test, and when do you test it?: Why the Hardest Part is Starting Early Enough

Wait until next week. We'll have a better sketch on a bigger napkin.

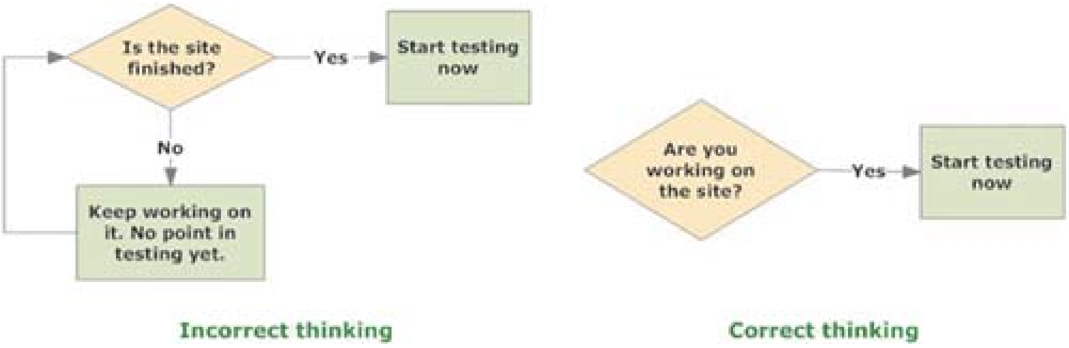
—WHAT MY CLIENTS ALWAYS SAY WHEN I TELL THEM I WANT TO SEE THE DESIGN THEY'VE SKETCHED ON A NAPKIN

It's not hard to understand: If you're going to watch people try using what you're building, you have to have something for them to use. This means you have to decide what you're going to be testing each month.

People tend to think that you can't start testing until you have something that actually works—if not the finished product, then at least a functioning prototype.

But if there's one thing that usability professionals agree on, it's that you want to start testing as early as possible.

[View full size image]



They know from experience that it's possible to detect serious usability problems very early in the development process, even if you have very little to show.

And they also know that it's usually far easier and less costly in the long run if you can fix usability problems early, before you've started building out the site with the problems embedded in it. Sometimes major problems that are detected too late can't be corrected at all. The worst practice is the most common one: waiting to test until the site is done and ready to launch.

Unfortunately, professionals also know that people resist the idea of testing early. Some common reasons:

- "We don't have enough done yet." After all, if it doesn't work, how can people try using it? In fact, it's never too early to start showing your design ideas to users, beginning with your first rough sketches.
- "It's too rough." Designers are often reluctant to show things that look unfinished. But users may actually feel freer to comment candidly on something that looks rough, since they know it's still subject to change.
- "Why waste people's time looking at something we know we're going to change?" During the

> design process, you always have a better version in your head than you've committed to code or paper. Yes, users will come across problems that you already know about, but there will also be surprises. In fact, you're mostly in it for the surprises: the things you didn't think of, because you're too close to it or because you don't understand your users as well as you think you do.

Here's the best advice I can give you about when to test:

Start earlier than you
think makes sense.

Your natural instinct will be to wait, which is the worst thing you can do. There's an inherent paradox: the worse shape it's in, the less you want to show it—and the more you can benefit if you do.

Throughout any project your team is going to be producing design artifacts: rough sketches, wireframes, page comps, working prototypes, and more. You can learn from testing all of these things, as well as testing your existing site and other people's sites.

In the rest of this chapter, I'm going to describe the different kinds of things you can test, how to test them, and what you get out of it.

Testing your existing site

If you already have a site and you're about to begin redesigning it, the obvious place to start is by testing your existing site.

HOW YOU TEST IT:

Follow the process spelled out in Chapters 5 through 9.

WHAT YOU GET OUT OF IT:

You'll learn a lot about what you're currently doing wrong so you'll know what to avoid as you redesign. You may even want to go ahead and fix some of the worst problems you discover. Your redesign is going to take time, so why make your users suffer until it's done?

You'll also learn things you didn't know about how people actually use your site.

Testing other people's sites

Before you've designed anything of your own, you can get a lot of value out of testing other people's sites. They may belong to your competitors or they may just be sites that have the same kind of content or the same kinds of users as you. Or they may just be sites that have features you're thinking of implementing.

Other people's sites are an underutilized resource. I always like to say that someone has gone to the trouble of building a full-scale working prototype of a design approach to the same problems you're trying to solve, and then they've left it lying around for you to use.

Most people overlook this opportunity, but it can save you an enormous amount of work. If you're building a travel site, for instance, think how much you could learn by watching people book trips on other travel sites.

HOW YOU TEST IT:

Follow the process spelled out in Chapters 5 through 9.

Give people the key tasks you test on your site. You may want to have each user do the same tasks on two or three competitors' sites.

But at the debriefing (Chapter 10), instead of determining the worst problems (since you're obviously not going to fix them), the team should have lunch and discuss what worked well and what didn't and what lessons can be applied to your own project.

WHAT YOU GET OUT OF IT:

The purpose is to learn from what others have done: what works and what doesn't.

As you might imagine, testing other people's sites has great appeal to marketing and management: they're always curious about what the competition is doing. It's a great way to get them to come and watch tests—and get hooked on the process.

Doing a round of testing on other people's sites can also be a good way to get your feet wet without any pressure. People aren't going to be defensive because it's not their stuff being tested.

Testing the sketch on the napkin

During the early planning stages of any project, you're likely to have some rough sketches or concept drawings, what I usually refer to as the "sketch on a napkin." (It may even literally be a sketch on a napkin or a placemat.) For a Web site, you might have a sketch of a new Home page or a product page, for instance.

It's always worth testing the sketch on the napkin.

HOW YOU TEST IT:

Napkin tests aren't full tests; they're like the Home page tour you saw me do in the demo test (see page 21). Each one takes less than five minutes. You can do napkin tests using friends, neighbors, or anyone you run into, or you can do them where your actual users gather, like a trade show or a user group meeting.

Here's how you do it:

1. Approach almost anyone.
2. Say, "Can you do me a favor? Take a look at this?"
3. Hand them the napkin. (It could be a nice neat drawing, or it could actually be a sketch on a napkin.)
4. Say, "Can you tell me what you make of this? What do you think this is supposed to be?"
   Note that you're not asking for their opinion ("Do you like this?") or their feedback ("What do you think of this?"). You're asking them to look at the sketch and try to figure out what the thing *is*.
5. Listen carefully. They'll probably say something like "Well, it looks like a Home page for a site, and it looks like you're trying to sell ____. And these things over here are your featured products. And it says 'Store' up here, so I guess I could order things online. I'm not sure what this category 'Incentives' means, though."
   If you want, you can ask a few probing questions, like "What do you think 'Incentives' might mean?"

If what they describe is what you were aiming for, get a bigger napkin and keep drawing. Usually, though, there will be something about the sketch that doesn't make sense to them, or something that they interpret very differently from what you expect, and you've learned something important without building anything—something you can now fix before you go any further.

WHAT YOU GET OUT OF IT:

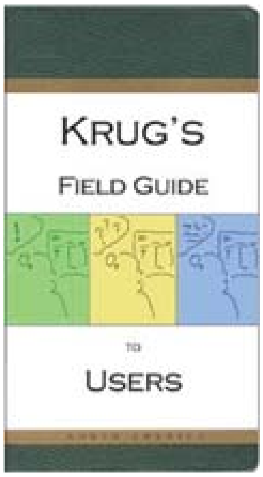
You'll learn whether your concept is easy to understand—whether people "get it." They'll either confirm that you're on the right track or point out basic problems that you can then deal with early in the process.

I'll give you a personal example. For a long time (several years, actually) I wanted to call this book

Krug's Field Guide to Users. The whole design of the book was going to be like a bird watching book: the same size and shape, and the same look and feel.

I thought it was a great idea. No, that's not quite right: I thought it was a *fabulous* idea. I loved it. Just thinking about it made me happy. I kept a rough version of the cover on the wall near my desk for inspiration.[1]

[1] *Actually, there was one title I would have liked even more:* The Junior Woodchucks Guidebook *(the pocket-size volume always carried by Donald Duck's nephews Huey, Dewey, and Louie that contained information and advice on every possible subject). But I knew that the intellectual property folks at the Disney Corporation wouldn't have been pleased.*



Then I did a foolish thing: I followed my own advice and tested it. The results were unanimous:

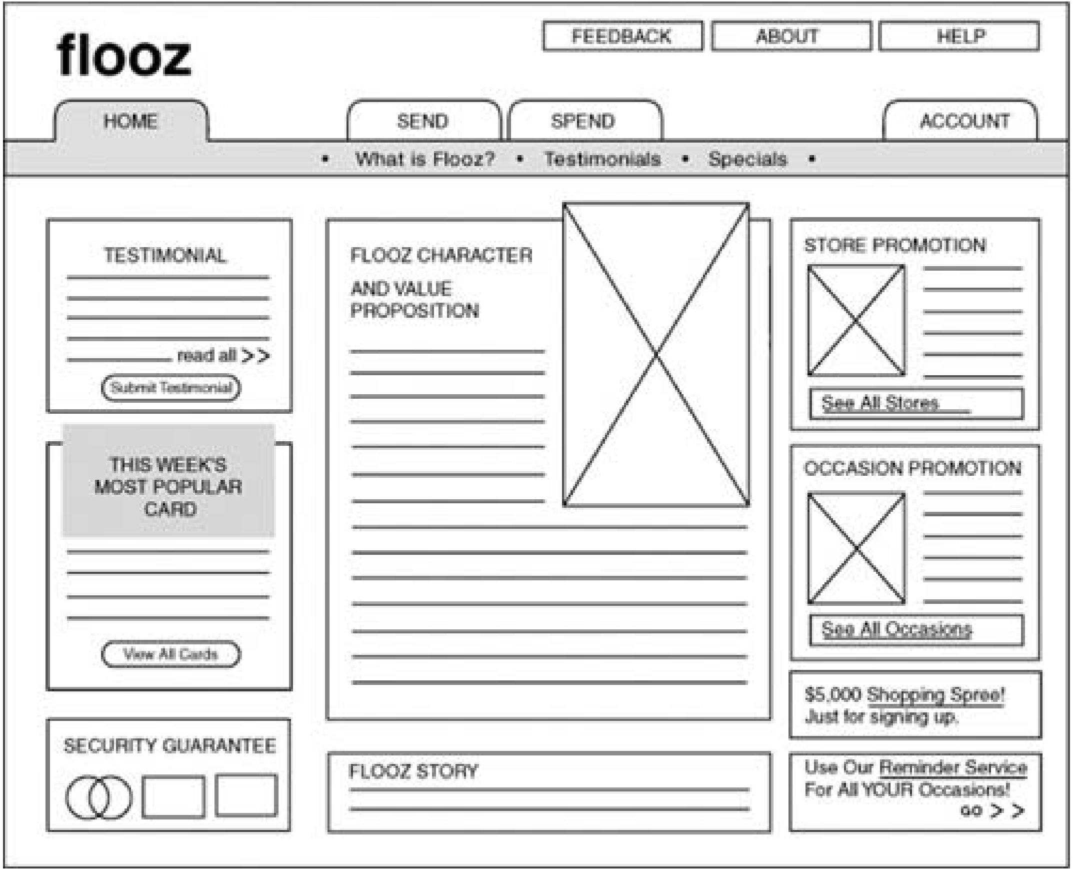
- Everybody I showed it to "got it" that it was supposed to be like a bird watching book. They all thought that it was a "neat" idea.
- They all thought that it would be a book about all the different kinds of Web users. When I told them that it would actually be about usability *testing*, they all went, "Oh...." They weren't upset that I was writing a book about testing. It just wasn't what the cover would have led to them to expect.

I couldn't see it because I was too close to it. I knew how it was supposed to work.

Testing wireframes

After sketches, the usual next step in Web design is creating wireframes.

[View full size image]



A wireframe is essentially a schematic diagram of a page. Typically, it shows where different kinds of content will go, the relative prominence of things like headings, and the navigation devices like menus and search.

HOW YOU TEST IT:

You test a wireframe by making up tasks, usually all related to navigation: "How would you find ______?" "What would you expect to see when you click on this link?"

Wireframe tests won't take very long because there's not a lot people can do with them. You'll usually do them in a session which includes testing of other things, like your existing site or other people's sites.

WHAT YOU GET OUT OF IT:

The main thing you're testing is your categorization scheme and naming: Are things where people expect to find them? Do the category names you're using make sense? Is it clear how the navigation is supposed to work? You may find, for instance, that you've organized your site according to your org chart and users don't think that way.

Testing page designs

Typically, a Web site has a few unique pages (like the Home page) and a series of templates (like section front pages, article pages, and product pages) that are repeated throughout the site with different content. The next stage after wireframes is usually creating visual treatments (or “comps”) of these different types of pages. Where wireframes focus on interaction, comps focus on the visual design.

HOW YOU TEST IT:

Starting with the Home pages, you lead them by the hand through comps and ask them to do a narrative (page 75) of each one.

WHAT YOU GET OUT OF IT:

The purpose is to try to see if the visual design has introduced any usability issues. Can people figure out how each page is supposed to “work”?

Testing working prototypes and beyond

For the rest of the project, you're going to have working pieces of the site available to test, ranging from prototypes to completed sections to the finished site.

HOW YOU TEST IT:

Follow the process spelled out in Chapters 5 through 9.

WHAT YOU GET OUT OF IT:

All the insights you need to improve your site.