TER

4

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

WHY THE HARDEST PART IS STARTING EARLY ENOUGH

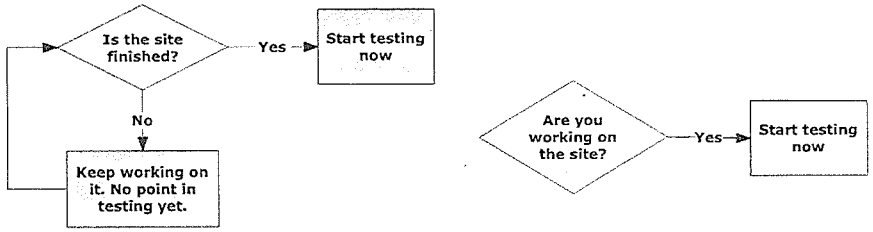
Wait until next week.
We'll have a better sketch on a bigger napkin.

—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.



Incorrect thinking

Correct thinking

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?"

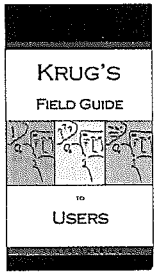
Note that you're not asking for their opinion ("Do you like this?") or their feedback ("What do you think of this?"). You're asking them to look at the sketch and try to figure out what the thing *is*.

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.*

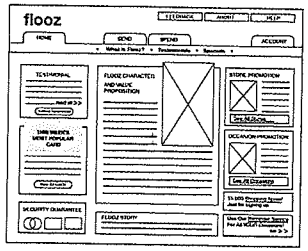
Then I did a foolish thing: I followed my own advice and tested it. The results were unanimous:

- 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.



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:

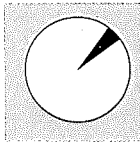
Follow the process spelled out in Chapters 5 through 9.

WHAT YOU GET OUT OF IT:

All the insights you need to improve your site.

Feel free to ask them follow-up questions. I usually ask at least one question about their job, like what their title means or what their company does. And if you don't understand something they say ("We broker irreducible energy credits"), don't pretend that you do. Ask them to explain it.

- **Get the information you need to grade on a curve.** By the time the participant has answered these questions, you'll have a pretty clear idea of (a) what they do for a living and (b) how computer-savvy and Web-savvy they are. This—plus the sense of the extent of their domain knowledge that you get from their reaction to the Home page (coming up next)—is usually all you need to decide how this person compares to your target audience.



The Home Page Tour
3 Minutes

I always start a Web site test by having the participant look around the Home page and tell me briefly what they make of it.

> OK, great. We're done with the questions, and we can start looking at things.
>
> First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.
>
> You can scroll if you want to, but don't click on anything yet.

The point is to see if the nature of the site is clear: Can users figure out what this thing *is*? As I'll explain later,[6] more often than you'd expect, the answer is a surprising—and revealing—"No."

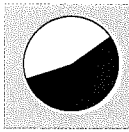
Having them do this "narrative" also gives you some idea of what they already know about the site, the organization behind it, and the subject matter—their domain knowledge.

Note that you're not asking them for their *opinion* of the Home page. The script doesn't say "Look around the Home page and tell me what you *think of it*." The instruction is carefully worded so they actually have a specific task to do: Figure out what this site *is*. This is a realistic (and important) task, one that people do on their own whenever they come to a new site. You're just asking them to verbalize it.

And it doesn't use much time, since most people will run out of things to say in two or three minutes. You don't want to let it go on longer than three minutes anyway.

The script tells them that they can scroll, but they shouldn't click on anything yet. If they click on a link anyway—and some people will—step in right away and ask them to go back to the Home page. Just say, "For right now, I just want you to stick to the Home page. Could you go back?"

[6] *The Big Bang Theory of Web Usability on page 122.*



The Tasks
35 Minutes

The tasks are the "meat" of the test.

At the beginning of each task, you'll hand the participant a copy of the scenario and then read it aloud, word-for-word, exactly the way you wrote it.

> Now I'm going to ask you to try doing some specific tasks. I'm going to read each one out loud and give you a printed copy.
>
> I'm also going to ask you to do these tasks without using Search. We'll learn a lot more about how well the site works that way.
>
> And again, as much as possible, it will help us if you can try to think out loud as you go along.

Why not just let the participant read it? If you do, some people won't read it carefully enough and they'll end up wasting time based on some misunderstanding of the task. If you read it to them, at least you'll know that they've heard every word of it.

Once they start a task, try not to interrupt any more than necessary. Basically, just keep them focused on the task and thinking aloud until it's time to move on to the next task.

How do you decide it's time to move on?

- **Have they completed the task?** If they have, hand them the next scenario and start the next task. (If they *think* they have, but haven't really, you can ask if they could try doing it again another way, which will usually lead them to realize their mistake.)

- **Is the participant miserable?** It's not unusual for the participant to experience a wide range of feelings while doing the tasks. With apologies to Elisabeth Kübler-Ross:

  Making the user miserable is overrated. I actually think you learn less from a miserable user. (As someone[7] has pointed out, it's not a crash test; you don't have to actually destroy the car to see the problems.)

  You don't need to stop at the first sign of a struggle, but if there *is* a struggle, you need to start thinking "Is this worth it? Is it causing the participant too much discomfort?" Always err on the side of the participant's feelings.

- **How much time do you have left, and is it important to get on to some other tasks?** Unless this is the last task in the session, you always want to be keeping an eye on the clock.

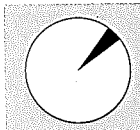
- **Are you still learning something?** My rule of thumb is this: when it starts to feel like you're not likely to learn anything more by continuing, let them continue a little bit longer and then move on. About half the time something useful will happen in this "overtime."

If the participant hasn't finished the task but you've decided to move on, just wait for a natural pause and then say something like "That's great. Very helpful. I want to move us along, since we've got more to do." (Note the use of "us" and "we" to avoid any suggestion that you're doing this because of some failure on the participant's part.)

[7] *...who I hope will identify himself, so I can give him credit...*

Feel free to ask them follow-up questions. I usually ask at least one question about their job, like what their title means or what their company does. And if you don't understand something they say ("We broker irreducible energy credits"), don't pretend that you do. Ask them to explain it.

- **Get the information you need to grade on a curve.** By the time the participant has answered these questions, you'll have a pretty clear idea of (a) what they do for a living and (b) how computer-savvy and Web-savvy they are. This—plus the sense of the extent of their domain knowledge that you get from their reaction to the Home page (coming up next)—is usually all you need to decide how this person compares to your target audience.



The Home Page Tour
3 Minutes

I always start a Web site test by having the participant look around the Home page and tell me briefly what they make of it.

> OK, great. We're done with the questions, and we can start looking at things.
>
> First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.
>
> You can scroll if you want to, but don't click on anything yet.

The point is to see if the nature of the site is clear: Can users figure out what this thing *is*? As I'll explain later,[6] more often than you'd expect, the answer is a surprising—and revealing—"No."

Having them do this "narrative" also gives you some idea of what they already know about the site, the organization behind it, and the subject matter—their domain knowledge.

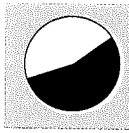
Note that you're not asking them for their *opinion* of the Home page. The script doesn't say "Look around the Home page and tell me what you *think of it*." The instruction is carefully worded so they actually have a specific task to do: Figure out what this site *is*. This is a realistic (and important) task, one that people do on their own whenever they come to a new site. You're just asking them to verbalize it.

And it doesn't use much time, since most people will run out of things to say in two or three minutes. You don't want to let it go on longer than three minutes anyway.

The script tells them that they can scroll, but they shouldn't click on anything yet. If they click on a link anyway—and some people will—step in right away and ask them to go back to the Home page. Just say, "For right now, I just want you to stick to the Home page. Could you go back?"

---

[6] *The Big Bang Theory of Web Usability on page 122.*



The Tasks
35 Minutes

The tasks are the "meat" of the test.

At the beginning of each task, you'll hand the participant a copy of the scenario and then read it aloud, word-for-word, exactly the way you wrote it.

> Now I'm going to ask you to try doing some specific tasks. I'm going to read each one out loud and give you a printed copy.
>
> I'm also going to ask you to do these tasks without using Search. We'll learn a lot more about how well the site works that way.
>
> And again, as much as possible, it will help us if you can try to think out loud as you go along.

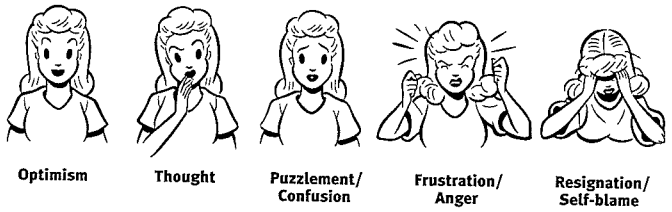
Why not just let the participant read it? If you do, some people won't read it carefully enough and they'll end up wasting time based on some misunderstanding of the task. If you read it to them, at least you'll know that they've heard every word of it.

Once they start a task, try not to interrupt any more than necessary. Basically, just keep them focused on the task and thinking aloud until it's time to move on to the next task.

How do you decide it's time to move on?

- **Have they completed the task?** If they have, hand them the next scenario and start the next task. (If they *think* they have, but haven't really, you can ask if they could try doing it again another way, which will usually lead them to realize their mistake.)

- **Is the participant miserable?** It's not unusual for the participant to experience a wide range of feelings while doing the tasks. With apologies to Elisabeth Kübler-Ross:



  Making the user miserable is overrated. I actually think you learn less from a miserable user. (As someone[7] has pointed out, it's not a crash test; you don't have to actually destroy the car to see the problems.)

  You don't need to stop at the first sign of a struggle, but if there *is* a struggle, you need to start thinking "Is this worth it? Is it causing the participant too much discomfort?" Always err on the side of the participant's feelings.

- **How much time do you have left, and is it important to get on to some other tasks?** Unless this is the last task in the session, you always want to be keeping an eye on the clock.

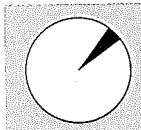
- **Are you still learning something?** My rule of thumb is this: when it starts to feel like you're not likely to learn anything more by continuing, let them continue a little bit longer and then move on. About half the time something useful will happen in this "overtime."

If the participant hasn't finished the task but you've decided to move on, just wait for a natural pause and then say something like "That's great. Very helpful. I want to move us along, since we've got more to do." (Note the use of "us" and "we" to avoid any suggestion that you're doing this because of some failure on the participant's part.)

[7] *...who I hope will identify himself, so I can give him credit...*

Feel free to ask them follow-up questions. I usually ask at least one question about their job, like what their title means or what their company does. And if you don't understand something they say ("We broker irreducible energy credits"), don't pretend that you do. Ask them to explain it.

- **Get the information you need to grade on a curve.** By the time the participant has answered these questions, you'll have a pretty clear idea of (a) what they do for a living and (b) how computer-savvy and Web-savvy they are. This—plus the sense of the extent of their domain knowledge that you get from their reaction to the Home page (coming up next)—is usually all you need to decide how this person compares to your target audience.



The Home Page Tour
3 Minutes

I always start a Web site test by having the participant look around the Home page and tell me briefly what they make of it.

> OK, great. We're done with the questions, and we can start looking at things.
>
> First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.
>
> You can scroll if you want to, but don't click on anything yet.

The point is to see if the nature of the site is clear: Can users figure out what this thing *is*? As I'll explain later,[6] more often than you'd expect, the answer is a surprising—and revealing—"No."

Having them do this "narrative" also gives you some idea of what they already know about the site, the organization behind it, and the subject matter—their domain knowledge.

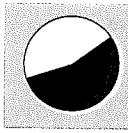
Note that you're not asking them for their *opinion* of the Home page. The script doesn't say "Look around the Home page and tell me what you *think of it*." The instruction is carefully worded so they actually have a specific task to do: Figure out what this site *is*. This is a realistic (and important) task, one that people do on their own whenever they come to a new site. You're just asking them to verbalize it.

And it doesn't use much time, since most people will run out of things to say in two or three minutes. You don't want to let it go on longer than three minutes anyway.

The script tells them that they can scroll, but they shouldn't click on anything yet. If they click on a link anyway—and some people will—step in right away and ask them to go back to the Home page. Just say, "For right now, I just want you to stick to the Home page. Could you go back?"

---

[6] *The Big Bang Theory of Web Usability on page 122.*



The Tasks
35 Minutes

The tasks are the "meat" of the test.

At the beginning of each task, you'll hand the participant a copy of the scenario and then read it aloud, word-for-word, exactly the way you wrote it.

> Now I'm going to ask you to try doing some specific tasks. I'm going to read each one out loud and give you a printed copy.
>
> I'm also going to ask you to do these tasks without using Search. We'll learn a lot more about how well the site works that way.
>
> And again, as much as possible, it will help us if you can try to think out loud as you go along.

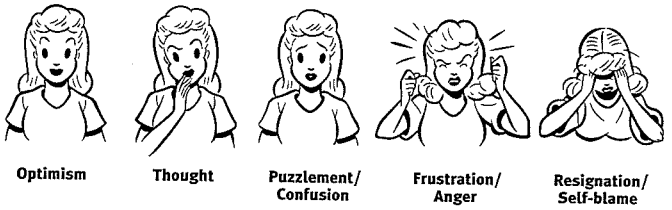
Why not just let the participant read it? If you do, some people won't read it carefully enough and they'll end up wasting time based on some misunderstanding of the task. If you read it to them, at least you'll know that they've heard every word of it.

Once they start a task, try not to interrupt any more than necessary. Basically, just keep them focused on the task and thinking aloud until it's time to move on to the next task.

How do you decide it's time to move on?

- **Have they completed the task?** If they have, hand them the next scenario and start the next task. (If they *think* they have, but haven't really, you can ask if they could try doing it again another way, which will usually lead them to realize their mistake.)

- **Is the participant miserable?** It's not unusual for the participant to experience a wide range of feelings while doing the tasks. With apologies to Elisabeth Kübler-Ross:



  Making the user miserable is overrated. I actually think you learn less from a miserable user. (As someone[7] has pointed out, it's not a crash test; you don't have to actually destroy the car to see the problems.)

  You don't need to stop at the first sign of a struggle, but if there *is* a struggle, you need to start thinking "Is this worth it? Is it causing the participant too much discomfort?" Always err on the side of the participant's feelings.

- **How much time do you have left, and is it important to get on to some other tasks?** Unless this is the last task in the session, you always want to be keeping an eye on the clock.

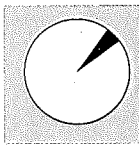
- **Are you still learning something?** My rule of thumb is this: when it starts to feel like you're not likely to learn anything more by continuing, let them continue a little bit longer and then move on. About half the time something useful will happen in this "overtime."

If the participant hasn't finished the task but you've decided to move on, just wait for a natural pause and then say something like "That's great. Very helpful. I want to move us along, since we've got more to do." (Note the use of "us" and "we" to avoid any suggestion that you're doing this because of some failure on the participant's part.)

[7] *...who I hope will identify himself, so I can give him credit...*

Feel free to ask them follow-up questions. I usually ask at least one question about their job, like what their title means or what their company does. And if you don't understand something they say ("We broker irreducible energy credits"), don't pretend that you do. Ask them to explain it.

- **Get the information you need to grade on a curve.** By the time the participant has answered these questions, you'll have a pretty clear idea of (a) what they do for a living and (b) how computer-savvy and Web-savvy they are. This—plus the sense of the extent of their domain knowledge that you get from their reaction to the Home page (coming up next)—is usually all you need to decide how this person compares to your target audience.



The Home Page Tour
3 Minutes

I always start a Web site test by having the participant look around the Home page and tell me briefly what they make of it.

> OK, great. We're done with the questions, and we can start looking at things.
>
> First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.
>
> You can scroll if you want to, but don't click on anything yet.

The point is to see if the nature of the site is clear: Can users figure out what this thing *is*? As I'll explain later,[6] more often than you'd expect, the answer is a surprising—and revealing—"No."

Having them do this "narrative" also gives you some idea of what they already know about the site, the organization behind it, and the subject matter—their domain knowledge.

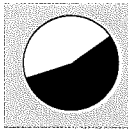
Note that you're not asking them for their *opinion* of the Home page. The script doesn't say "Look around the Home page and tell me what you *think of it*." The instruction is carefully worded so they actually have a specific task to do: Figure out what this site *is*. This is a realistic (and important) task, one that people do on their own whenever they come to a new site. You're just asking them to verbalize it.

And it doesn't use much time, since most people will run out of things to say in two or three minutes. You don't want to let it go on longer than three minutes anyway.

The script tells them that they can scroll, but they shouldn't click on anything yet. If they click on a link anyway—and some people will—step in right away and ask them to go back to the Home page. Just say, "For right now, I just want you to stick to the Home page. Could you go back?"

---

[6] *The Big Bang Theory of Web Usability on page 122.*



The Tasks
35 Minutes

The tasks are the "meat" of the test.

At the beginning of each task, you'll hand the participant a copy of the scenario and then read it aloud, word-for-word, exactly the way you wrote it.

> Now I'm going to ask you to try doing some specific tasks. I'm going to read each one out loud and give you a printed copy.
>
> I'm also going to ask you to do these tasks without using Search. We'll learn a lot more about how well the site works that way.
>
> And again, as much as possible, it will help us if you can try to think out loud as you go along.

Why not just let the participant read it? If you do, some people won't read it carefully enough and they'll end up wasting time based on some misunderstanding of the task. If you read it to them, at least you'll know that they've heard every word of it.

Once they start a task, try not to interrupt any more than necessary. Basically, just keep them focused on the task and thinking aloud until it's time to move on to the next task.

How do you decide it's time to move on?

- **Have they completed the task?** If they have, hand them the next scenario and start the next task. (If they *think* they have, but haven't really, you can ask if they could try doing it again another way, which will usually lead them to realize their mistake.)

- **Is the participant miserable?** It's not unusual for the participant to experience a wide range of feelings while doing the tasks. With apologies to Elisabeth Kübler-Ross:

  Making the user miserable is overrated. I actually think you learn less from a miserable user. (As someone[7] has pointed out, it's not a crash test; you don't have to actually destroy the car to see the problems.)

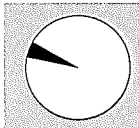
  You don't need to stop at the first sign of a struggle, but if there *is* a struggle, you need to start thinking "Is this worth it? Is it causing the participant too much discomfort?" Always err on the side of the participant's feelings.

- **How much time do you have left, and is it important to get on to some other tasks?** Unless this is the last task in the session, you always want to be keeping an eye on the clock.

- **Are you still learning something?** My rule of thumb is this: when it starts to feel like you're not likely to learn anything more by continuing, let them continue a little bit longer and then move on. About half the time something useful will happen in this "overtime."

If the participant hasn't finished the task but you've decided to move on, just wait for a natural pause and then say something like "That's great. Very helpful. I want to move us along, since we've got more to do." (Note the use of "us" and "we" to avoid any suggestion that you're doing this because of some failure on the participant's part.)

[7] *...who I hope will identify himself, so I can give him credit...*



Probing
5 Minutes

While the participant is doing the tasks, you'll inevitably notice things that you'd like to know more about.

> Thanks, that was very helpful.
>
> If you'll excuse me for a minute, I'm just going to see if the people on the team have any follow-up questions they'd like me to ask you.

But stopping to ask questions tends to interrupt the user's flow and train of thought and introduces the risk of your inadvertently giving "clues."

That's why you always want to leave some time at the end to go back and probe. It's your chance to make sure you understand what happened and to try to figure out—with the participant's help—why it happened.

While the participant is doing the tasks, you can always ask for minor clarifications ("Do you mean the ___ over here?"). But for anything deeper—the "Why do you think you did that?" kind of questions—you need to jot down a note to yourself ("Didn't notice left nav" or "Chose second link. Why?" for example) and save it for the probing section.

Before you start asking your own questions, call the observation room and ask your Hall Monitor if there's anything the observers would like you to follow up on. (Feel free to use your own judgment about how to use the time available for probing, though. You don't have to do everything they ask you to.)

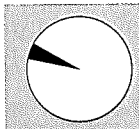
Typically, you'll want to ask the participants things like whether they noticed certain things and why they made particular choices. You can also ask them to try doing a task again another way, or from a different starting point.

If there are parts of the interface that you're interested in that they didn't get to in their travels, you can take them to specific pages ("I'd like you to go to the registration form") and ask them questions about them.

You may also want to follow up on any suggestions the participant made about features they think would be useful ("I wish there was a map to choose from

instead of an alphabetical list of states"). Occasionally these can turn out to be great ideas, but for the most part they're not.[8] Users aren't designers, and they don't always know what they need, or even what they really want. Usually, if you let them talk their idea through, they'll end up saying, "But I guess I really wouldn't use it. I'd probably keep doing it the way I do it now."

Sometimes, though, users *will* make brilliant suggestions. How can you tell? Don't worry; you'll know. If it's really a bright idea, a light bulb will go off over your head and the heads of everyone in the observation room. People will say things like "Why on earth didn't we think of that? It's so obvious."



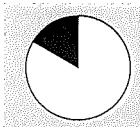
Wrapping Up
5 Minutes

Thank them, ask if they have any questions, pay them, and show them to the door. That's it.

> Do you have any questions for me, now that we're done?

At the end, I always like to say, "Thanks. That was exactly what we need. It's been very helpful."—even if things have gone badly. (Or especially when they've gone badly.)

[8] *Like the car designed by Homer Simpson with shag carpeting, two bubble domes, and three horns ("...because you can never find a horn when you're mad") that all play La Cucaracha, which ends up costing $82,000 to manufacture.*



Prepare For The Next Test
10 Minutes

Stop the screen recorder!

Save the recording!

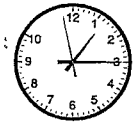
Clear the browser cache, history, and visited links

Open a "neutral" screen in the browser (e.g., Google)

Take time before the next session to jot down a few notes about things you observed

Notice that I've suggested that each test session last only 50 minutes, not a full hour. This is like the therapist's 50-minute hour—appointments are scheduled on the hour, but they last for 50 minutes—and it's done for the same reason. To get the most out of each session, you need some time between tests to clear your head, gather your thoughts, and perhaps fit in a bio-break.

Obviously this means that you only have 50 minutes for testing. If you want to do longer sessions, you're going to have to get a little funky with your start times. But always try to leave at least 10–15 minutes of down time between sessions. Don't make the break too long, though, because observers will end up drifting away to take care of "just one thing" and not come back.



During the break, you should

- **Make a few notes.** It will all run together, even with three tests.
- **Reset the computer.** You want to restore everything to the state it was in before the test. Reload your sample data and clear your browsing history.

- **Consider making adjustments.** Based on what you've seen in the previous session, you may decide to make changes to the test on the fly. For instance, if the first participant can't complete a task and the reason is obvious, you can modify the task—or even skip it—for the remaining participants. You may even want to implement a quick fix to what you're testing if it's something you can do by making a simple change to a style sheet or rewording a heading.

Freud would be proud of you

Ever since I started doing usability testing twenty years ago, I've been struck by how many of the things a facilitator does with participants are just like the things a therapist does with clients. For instance:

- **You're trying to get them to externalize their thought process.** You want to hear what they're thinking so you can understand what's confusing and troubling them. Your primary job is to keep 'em talking.
- **You're trying not to influence them.** Like a therapist, you need to remain neutral. You can't tell them what to do; they need to figure it out for themselves.
- **You say the same few things over and over.** Many of the phrases you'll use are the same ones therapists use.
- **You have ethical responsibilities.**

Keep 'em talking

You'll find that some participants will think aloud with only an occasional reminder. For the people who tend to forget to verbalize their thoughts, though, you have to decide how often you should prompt them.

I used to think that it was a function of how long they'd been quiet: if they hadn't said anything for 20 seconds (or 30, or 40—I was never quite sure what the right number was), then you'd ask what they were thinking. But I finally realized that it's something else:

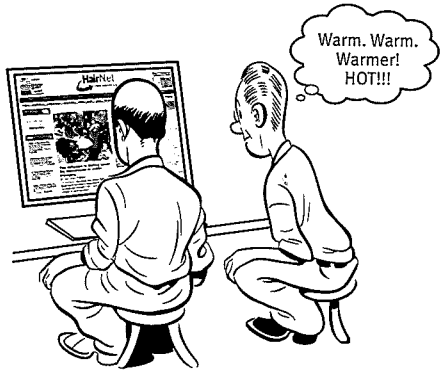
If you're not entirely sure you know what the user is thinking, ask.

Most of the time when the user is quiet, you'll still have a pretty good idea of what they're thinking. For instance if it's obvious that someone is reading something, you should just let them read. If they're making progress along a path that makes sense to you and they don't seem at all confused or hesitant, let them keep going. But as soon as you lose the feeling that you're certain you know what they're thinking, it's time to ask.

And you don't have to worry about it getting annoying. It turns out you can say "What are you thinking?" dozens of times in a test and participants won't even be aware of it. And if *you* get bored saying it, you can mix it up with "What are you looking at?" and "What are you doing?"—both of which have about the same effect.

Stay neutral

Like a therapist, one of the hardest parts of your job as facilitator is staying neutral: you don't want to influence the participants.



The worst case is when the facilitator is actively trying to advance a personal agenda, either consciously or unconsciously. For instance, you may want to see the thing you're testing succeed because you had a hand in designing it, or you may want to see it fail because you've thought all along it was a bad idea.

As facilitator, you have a responsibility to be aware of your biases and scrupulously steer clear of influencing what happens during the testing. If you don't, people will notice and your testing will lose its credibility.

But even if you don't have a personal agenda, you still have to do everything you can to avoid influencing the participant:

- You can't tell them what to do or give them clues—even subtle ones. When the participant is struggling, you'll want to help, but you need to resist the temptation.

- You can't answer their questions. You'll have to answer most questions with a question, like "What do *you* think?"
- You shouldn't express your own opinions ("That's a great feature"), or even agree with theirs ("Yeah, that *is* a great feature").
- You need to try to maintain a poker face, not giving any sign that you're particularly pleased or displeased with what's happening. (I think it's probably best to seem consistently *somewhat* pleased throughout—conveying the sense that the test is going well and you're getting what you need.)

"Things a therapist would say"

While the participant is doing the tasks, to maintain your neutrality you're going to be saying the same few things over and over. Here's a handy chart:

WHEN THIS HAPPENS:	SAY THIS:
You're not absolutely sure you know what the participant is thinking.	"What are you thinking?" "What are you looking at?" "What are you doing now?"
Something happens that seems to surprise them. For instance, they click on a link and say "Oh" or "Hmmm" when the new page appears.	"Is that what you expected to happen?"
The participant is trying to get you to give him a clue. ("Should I use the _____?")	"What would you do if you were at home?" (Wait for answer.) "Then why don't you go ahead and try that?" "What would you do if I wasn't here?" "I'd like you to do whatever you'd normally do."
The participant makes a comment, and you're not sure what triggered it.	"Was there something in particular that made you think that?"
The participant suggests concern that he's not giving you what you need.	"No, this is very helpful." "This is exactly what we need."

The participant asks you to explain how something works or is supposed to work (e.g., "Do these support requests get answered overnight?").	"What do you think?" "How do you think it would work?" "I can't answer that right now, because we need to know what you would do when you don't have somebody around to answer questions for you. But if you still want to know when we're done, I'll be glad to answer it then."
The participant seems to have wandered away from the task.	"What are you trying to do now?"

There are also three other kinds of things you can say:

- **Acknowledgment tokens.** You can say things like "uh huh," "OK," and "mm hmm" as often as you think necessary. These signal that you're taking in what the participant is saying and you'd like them to continue along the same lines. Note that they're meant to indicate that you understand what the participant is saying, not that you necessarily agree with it. It's "OK." Not "OK!!!"
- **Paraphrasing.** Sometimes it helps to give a little summary of what the participant just said ("So you're saying that the boxes on the bottom are hard to read?") to make sure that you've heard and understood correctly.
- **Clarifying for observers.** If the user makes a vague reference to something on the screen, you may want to do a little bit of narration to make it easier for the observers to follow the action. For instance, when the user says "I love this," you can say, "The list over here on the right?" (Since you're sitting next to the participant, you sometimes have a better sense of what they're looking at.)

Ethical considerations

There's one final thing you have in common with a therapist: you have an ethical responsibility to your participants. Like anything to do with ethics, this responsibility can be complicated, but I like to think it boils down to this:

Participants should leave the room in no worse shape than they entered.

For the most part usability testing tends to be very benign. You're not attaching electrodes to anyone, and unless you're a closet sociopath, I don't think you're likely to cause anyone serious emotional damage. I assume you're going to treat them with respect, empathy, and consideration of their feelings, even if they turn out to be a pain in the neck. (Perhaps *especially* if they turn out to be a pain in the neck.) In other words, you're going to behave like a decent human being.

The participant always has the right to stop the test and leave at any time without penalty. (You still pay them.) You should work to make the test as comfortable, unintimidating, and stress-free as possible, keep a close eye on the participant's comfort level, and be very gracious and agreeable if they *do* want to stop. In some rare cases, *you'll* ask *them* if they'd like to stop.

You also have a responsibility to protect the participants' privacy. One of the best ways to do this is to avoid using identifying information. There's no need to use their last names in the tests or recordings, and you're not going to record their faces.

You need to keep the recordings under your personal control and erase them as soon as they're no longer needed. If you're going to distribute clips within your organization, each one should begin with a scary-sounding FBI-style warning not to redistribute it, and you should redact any personal information like telephone or credit card numbers. (It's fairly easy to cover things up with the editing features of Camtasia.) And if someone makes a particularly indiscreet (or even incriminating) statement, you should delete that portion of the recording.[9] I would also never distribute clips of employees who were participants because it may put them in an awkward position.

If you're in an academic setting, you may be required to get approval of your entire test plan (including the script and an informed consent agreement) from your Institutional Review Board (IRB) to ensure that it meets your institution's ethical standards. But you can probably make a very good case that informal usability tests like this are not the kind of study that your IRB has to oversee. (People have managed to get this kind of exemption in the past.)

[9] *Carolyn Snyder has talked about doing this when a participant mentioned smoking pot, for instance. I was once testing a site with some college students (at a Catholic university, no less) and asked casually what kinds of sites a participant used. "Well, there's porn..." he began. I left this clip out of my presentation.*