Chapter 9

Usability Testing with a Paper Prototype

The previous chapter covered the basics of usability testing without regard to whether the interface is being tested on a computer or as a paper prototype. This chapter discusses the variations in usability testing methodology that are specific to the medium of paper.

The Test Facility

It's not necessary to have a usability lab to test a paper prototype—I've conducted more than 90% of my usability tests in my clients' conference rooms. All you really need is a room with a door that can be closed (to minimize distractions) and a table large enough for the users and the Computer to interact with the prototype.

Tip: You might not want to reserve your best conference room for usability testing—I've learned that the fancier the conference room, the wider the table. Very wide tables make it hard for the Computer and users to work on a prototype that's between them.

On the other hand, there is nothing wrong with testing a paper prototype in a usability lab if you happen to have one. The biggest problem I've found with usability labs is that some are optimized for computer-based testing—the rooms are small, or there's restrictive furniture (such as a kneehole desk) that makes it hard for several people to gather around. If your lab lacks a large work surface, consider getting a couple of inexpensive folding tables to use when you're conducting paper prototype tests.

Seating

Obviously, you want the users to sit in front of the prototype. Put the users closest to the door so that they don't face an obstacle course on their way in and out. The Computer usually sits across from the users, which means that the Computer sees the prototype upside down—a drawback, but usually the Computer is familiar enough with the interface that it's not a problem. (And other prototype pieces waiting in the wings will be upside down to the users, and thus less likely to distract them.)

The facilitator can sit either next to the users or across the table, next to the Computer. It's easier for the facilitator to converse with users when they're across the table, but I also think there may be a subtle psychological benefit to having the facilitator literally "on the users' side." Although I usually sit next to the users, I think that both options work.

It's best to avoid having observers sit behind users because this can make some people uncomfortable. Instead, observers should sit around the table, where they can see the prototype. See Figure 9.1 for some sample seating arrangements. As explained in the next chapter, this isn't as distracting as it sounds if the observers are well behaved. A useful rule of thumb is that the less familiar an observer is with the prototype, the closer he or she gets to sit. This is exactly the opposite of what happens by default—someone unfamiliar with the interface often says, "Don't mind me; I'll just sit over here in the corner." But then that person can't follow what's going on, gets little value from observing, and doesn't bother to come again.

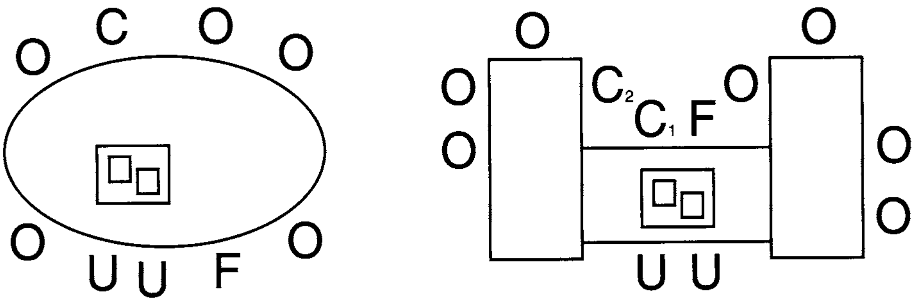


Figure 9.1 Some possible seating arrangements for a paper prototype test. The letters indicate the positions.

Videotaping

I first started conducting usability tests when I joined User Interface Engineering in 1993. Back then, we videotaped all usability tests—paper or otherwise—as a matter of course, and we gave our clients copies of the tapes. But in recent years I have almost completely stopped videotaping usability tests because most tapes of paper prototype tests simply aren't used.

Whenever I cross paths with a former client, I ask what they did with the tapes from the usability study we conducted. Usually, the person looks slightly embarrassed and admits that the test tapes were put on a shelf and have been gathering dust ever since. I estimate that at least 90% of the usability test videotapes I've made have never been watched.

Reasons to Skip the Videotaping

Videotaping a paper prototype usability test is less useful than taping a computer-based test, for several reasons:

- **The action is slowed down.** On a computer, users may perform keyboard or mouse actions faster than the eye can follow, but with a paper prototype the speed is limited by how fast the human Computer can do things. The Computer can't miss the action because he or she controls it.
- **The interface changes frequently.** Many paper prototypes are revised after each usability test, and they can evolve quite rapidly. In a fast-moving development environment, a videotape that shows problems from 2 weeks ago may no longer be relevant.
- **There is less need for analysis.** Analysis of usability test data always has a point of diminishing returns. With paper prototypes, this point comes sooner than with software, again because the prototype tends to change rapidly—some statistics might be obsolete before they're even calculated. A useful question to ask yourself is, "Will the time needed to review that tape provide more value than something else I could be doing instead?" Perhaps the answer is yes if there's an important but subtle issue you're still trying to piece together, but

often the time would be better spent revising the interface or conducting an additional test.

- **It's hard to read the prototype screens.** With inexpensive video equipment, the picture may not be good enough to read the prototype screens and sometimes screens with a lot of white space can confuse a camera's automatic focus. Although better equipment might solve the problem, I'd go that route only if it wasn't physically possible for observers to come to the test.

Reasons to Videotape

On the other hand, here are some reasons why you might decide to videotape your usability tests.

- **Not enough observers.** As long as there are at least two or three good observers taking notes, they should be able to capture many of the issues. But if your team is very small, the observers are inexperienced, or you're worried that you might miss something important, a videotape can act as your safety net.
- **Remote observers.** Although I advocate having all the observers in the same room, this may not always be desirable or practical. With observers in another room, a video camera is usually needed to relay the action because the prototype may be difficult to see through a one-way mirror. Because you're already dealing with a video camera, you might as well pop in a tape.
- **Need for a highlight tape.** Highlight tapes can be valuable to communicate the problems you found to a large number of people or important stakeholders. Seeing the users get confused and voice their concerns has a lot more impact than simply hearing someone describe what happened. But highlight tapes can be time-consuming to make unless you have good editing equipment and know how to use it—consider whether a few pithy user quotes might do the trick. (As long as the user can't be identified by what they said, this use of test information is covered by the informed consent form.)
- **Need for a demonstration tape.** A demonstration tape is similar to a highlight tape except that its purpose is to illustrate the technique of paper prototyping instead of the findings from tests. Sometimes people need to see a brief clip of a paper prototype test to "get" how the users interact with the Computer and prototype. If you don't have a suitable test tape and the users' permission to use it, one alternative is to record a "dramatization" (as is sometimes done on

television) of real events, but using your co-workers as actors. This approach requires no editing because you prepare the script ahead of time, and you probably don't need more than a minute or two of tape to show how the method works. Naturally, you would make it clear that this was a reenactment, not a real test.

- **Just in case.** For whatever reason, someone may want you to tape the session "just in case." The main downside is that if you do a lot of testing, you'll eventually have a large library of obsolete tapes. One company I know of simply records over their oldest tapes.

How to Videotape a Paper Prototype

Having discouraged you from making unnecessary videotapes, it's okay to do so if you have a use for them. Naturally, an essential prerequisite is to obtain the users' written consent, including an explanation of the purposes for which the tape can be used—you don't want a user to turn on the TV and discover that he's in a commercial giving a testimonial for your product. So think about the possible reasons you might want to use a tape: for just you, to verify something that happened? To show to the product team? To make a demo tape or a training video on test facilitation? Make your consent form as explicit as possible. If your consent form says that you'll use the tape only to verify what happened in the session and you decide later that it would be perfect for your presentation at the annual industry conference, you must contact the users and get their permission.

Here are some tips on what equipment to use and how to set it up.

- **Film from above.** I've made dozens of tapes by mounting a consumer-grade video recorder on a tripod behind the users. I place the tripod as close as practical, extend it fully, and angle the camera downward to focus on the prototype. Although this quick-and-dirty approach is good enough for many situations, it's inevitable that the camera's view of the prototype will be partially obstructed at times by the users. If you have a usability lab, a camera in the ceiling can capture the action. If you don't have a lab, look for a tripod that allows the camera to be mounted vertically above the table (Figure 9.2).
- **Determine the field of view.** You want to ensure that observers can read the screens, so the prototype should fill most of the camera's view. Be aware of what the camera can and cannot see. Tape the prototype background to the

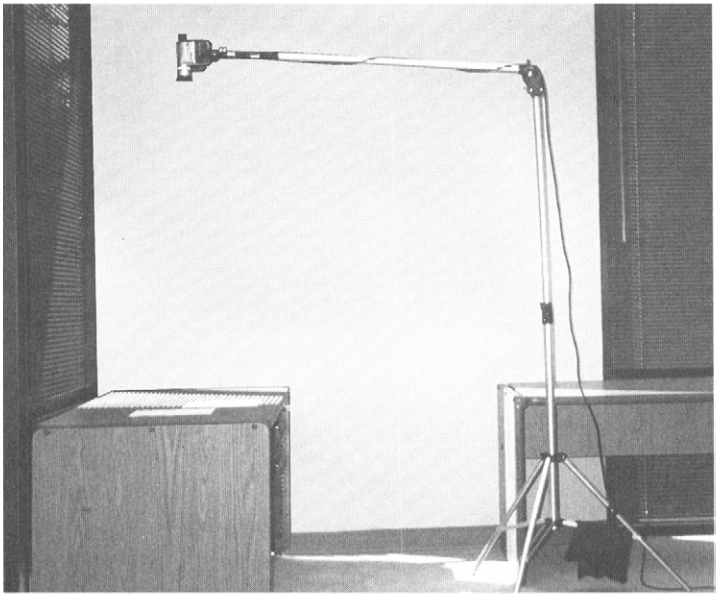


Figure 9.2 A special tripod allows you to position the camera directly above the prototype.

table to keep it in place during the test. Alternatively, you can outline the camera's field of view with masking tape and keep all the action inside this box. (A tip for those with fancy equipment: If you're using multiple cameras and a mixer to put an inset of the user's face in a corner of the recording, part of the prototype video won't be visible. Mark out that section on the prototype as a reminder not to let important action happen there.)

- **Use a trained camera operator if necessary.** If the camera is positioned and focused correctly, a human camera operator may not be necessary. However, if you have a lab technician who knows what he or she is doing, it's appropriate to make camera adjustments to help the viewers see better. But keep adjustments to a minimum. When I made my first few videotapes, my overzealousness in panning, zooming, and focusing made the resulting tapes a recipe for visually induced motion sickness.

- **Watch the wiring.** Run cables where the users won't trip over them. Ideally, everyone in the room should have a clear path to the door without having to avoid cables. (Back in the days of clip-on microphones, I once tripped over a cord and yanked the mic right off the user's collar. Neither of us was hurt, but it taught me a lesson.)

Preparing Users

So now you've got your room and equipment ready. This section describes how to prepare users for a usability test of a paper prototype. The goal is to have users feel prepared, comfortable, and in control of the session. I'm assuming that you are using co-discovery and in-room observers—if not, simply omit those instructions that do not apply to your situation.

Note: All the tables in this chapter are available as Word documents from *www.paperprototyping.com.*

During Recruitment

My advice on explaining paper prototyping during user recruitment is "Don't." Paper prototyping is an inherently visual and kinesthetic activity. Trying to explain it over the phone to a stranger (who probably knows little or nothing about usability testing) is like using mime to explain life insurance to a 10-year-old—you're using the wrong medium, and they lack the background. You'll get tongue-tied, the user will be confused, and you'll get distracted from the primary purpose of the recruitment process, which is to schedule participants who are suitable and willing.

Instead, gloss over paper prototyping by saying something like, "You'll be working with a prerelease version of the system." You want the user to understand the *goals* of the session—to find out where the interface works well and where it's confusing—but you don't need to describe your technology, or lack thereof. Here is an example of a paragraph that I gave to a market research firm that was recruiting participants to test a data security application:

> We're looking for participants for a usability study. In this study, you'd be working with a pre-release version of a web application intended for use by people who are responsible for data security and web servers. Our goal is to figure out how to make the interface more useful and user-friendly. We would ask you to perform some tasks with the application while members of the development team sit in the room, quietly observing and taking notes. We are testing the web application; we are not testing you! Depending on the session there may or may not be another participant working with you, and there will be a facilitator sitting next to you to assist you. Does this sound like something you'd be comfortable doing?

Upon Arrival

Once the users arrive, you want to prepare them for what they'll encounter when they enter the test room. Your first few minutes with the users sets the tone for the session, so strive to be professional, relaxed, and friendly. Table 9.1 shows the items you should cover in your introduction and examples of wording you might use.

In the Test Room

The observers are assembled in the room before the users come in, and the prototype is on the table, displaying the screen that we've decided to use as the starting point, even if it's just a blank background. Once the users are seated in front of the prototype, I ask the team members to introduce themselves first—this allows the users to look around the room and get familiar with the test setting before they are asked to speak. Observers should avoid giving much detail about themselves or what they're working on, especially things like "lead designer," which might sound intimidating. If in doubt, just names will suffice.

Then I introduce each user by first name (do whatever is appropriate in your culture) and have each one describe a bit about his or her background. Not only does this get the users comfortable speaking to the group, but the team can verify that the users meet the profile of the target audience for the interface. Whoever did the user recruitment should provide you with the users' answers to the screening questions, which can help you identify interesting aspects of the users' background. I usually ask two or three specific questions—in the absence of structure, some users will either say very little (bad) or will give their entire life history (worse). I also give the observers a chance to ask the users more about their background before we begin the test.

Before giving the users the first task, I explain what they're looking at and how we'd like them to interact with the paper prototype and Computer. I always end my introduction with the all-important, "We're testing the interface; we're not testing you." I say this to every user at least once before the test (even if they've participated in a test before), and repeat it if necessary. See Table 9.2 for details and examples of wording.

How Users React

No doubt about it—paper prototyping feels weird for the first few minutes. Actually, so does any kind of usability testing. From the users' perspective, it's like

Table 9.1 Pretest Briefing

Topic	*Checklist*	*Example Wording (i.e., Script)*
Greeting and introduction	◇ Welcome the users (hang up coats, offer a beverage, ask if the directions were okay). ◇ Introduce yourself and the company. ◇ Describe the interface being tested. ◇ Give users the informed consent form (ideally, they brought the copy you sent ahead of time). ◇ Explain the goals of the session. ◇ Introduce the notion of paper prototyping.	"Thank you for coming. I'm Carolyn Snyder. I'm an independent consultant, and I specialize in conducting sessions like this one. Here at [company] we're working on a product for [target market] that will help them to [basic functionality]. I'll go over the main points in this form we sent you. The purpose of today's session is for you to help us figure out how to make this interface more user-friendly before we finish developing it. But believe it or not, we aren't going to use a computer. As you'll see, we've actually created paper versions of the screens, and this guy named Carl will be playing the computer."
Their role	◇ If using co-discovery, introduce the two users. ◇ Explain what's expected of them. ◇ Remind them of their qualifications.	"Frank, this is Ernest. The two of you will be working together. We'll give you some tasks that we think are representative of what people might do in real life. [Give example.] Your job is to tell us what makes sense, what's confusing, whether it works the way you'd expect it to, etc. You are here because you know [area of expertise], so your perspective will help us make this product more useful."
Social concerns	◇ Explicitly mention in-room observers and/or videotaping. ◇ Explain that you're testing the interface, not them. ◇ Reassure users about what will happen if they encounter any difficulties.	"About half a dozen members of the development team will be sitting in the same room, observing quietly and taking notes. We're not going to be videotaping. Keep in mind that we're testing the interface—we're not testing you—so if you run into any problems it's not your fault and it means that there's something

Continued

Table 9.1 Pretest Briefing—cont'd

Topic	*Checklist*	*Example Wording (i.e., Script)*
	◇ Reiterate how valuable this is and how much you appreciate their help.	we need to change. I'll be sitting next to you, and I can help you if you want. We held our first session this morning, and we learned a lot; in fact, we've already made some changes. We really appreciate having you come and help us out."
Set expectations	◇ Acknowledge the unfinished nature of the prototype (avoid the temptation to apologize—present this as a benefit). ◇ Explain that the design will evolve. ◇ Explain that you will record their suggestions but don't promise to implement them (especially important if the user is a customer).	"The prototype still has some rough edges—we're still thinking through how it should work and some parts of it are incomplete. Before we cast it in concrete, we want to get some feedback about how well this design works. We're doing several sessions like this one, so it's likely that the final version of the interface will be different than what you see today. If you have suggestions we'll make note of them, although at this point it is premature to promise what we'll be able to include in the interface. When we get done with this series of sessions, we'll review everyone's feedback to help determine our priorities for the next release."
Paperwork and administrivia	◇ Get signature on informed consent form. ◇ Pay users (unless you have decided to pay them at the end). ◇ Escort them into test room.	"Do you have any questions about what we'll be doing today? If not, could I please get your signature on this form? And so I don't forget, I'm going to give you your payment now since you've already earned it by virtue of showing up on time. If you need to leave early for any reason, you're still entitled to keep it."

Table 9.2 Introducing the Test

Topic	*Checklist*	*Example Wording*
Introductions	◇ Direct users to the chairs in front of the prototype. ◇ Ask observers to introduce themselves. ◇ Introduce users. ◇ Have users answer 2–3 questions about their background.	"Frank, Ernest, please have a seat here. I'll let these folks introduce themselves." "Frank, you're a senior network administrator at MegaBank, right? Please tell us a little bit about your responsibilities, how many end users you support, and what kinds of data security products you've used." (Same for Ernest.) "Does anyone here have any other questions for Frank or Ernest before we get started?"
Paper prototype orientation	◇ Explain what they're looking at (usually the first screen of the site or application). ◇ (Optional) For users with limited computer experience, you might point out familiar elements such as the browser buttons, explaining that these things on their computer might look a little different, but they do the same thing.	"As I mentioned, here's the paper prototype of the Web application you'll be working with. We've assumed you typed [URL].com into the browser and this page came up. This is your starting point." (Optional) "You use AOL at home, right? These forward and back buttons here do the same thing as the buttons in AOL—they just look a little different. If you're not sure whether something is the same as what you're used to, just ask me."
Interactions with the prototype and Computer	◇ Introduce the Computer. ◇ Explain how to interact with the prototype (and each other if co-discovery). ◇ Discuss thinking aloud and asking questions. ◇ (Optional) Note that all interface functionality is available (e.g., drag & drop).	"Carl here will be playing the computer. Now Carl may seem like a pretty smart computer, but he has no speech recognition and no artificial intelligence. Since machines can't talk, he's not allowed to explain anything. If you want to do something, you'll need to interact with the prototype just as you would on a computer. Use your finger to click on

Continued

Table 9.2 Introducing the Test—cont'd

Topic	*Checklist*	*Example Wording*
	◇ (Optional) Point out any print manuals that may exist or mention that there's a help system.	links or buttons. These pieces of tape indicate places where you can type something in, and here's your keyboard (give pen). It's okay to write on this. The two of you are working together, so talk to each other as you go. It's fine if you disagree about something because there are different ways to use this. Please tell us what makes sense to you, what's confusing, and any questions that come to mind. Your questions are especially valuable, but I may not answer them right away because our goal is to change the interface so it answers them." (Optional) "Even though this is a paper prototype, assume you can do all the things you can do with a real computer, like drag & drop and right mouse menus. There's even a help system."
They are in charge	◇ Remind the users that you're testing the interface. ◇ Confirm ending time and that they can stop or take a break at any time.	"Remember that we're testing the interface—we're not testing you. We'll end promptly at 4:00, but if you need to stop or take a break before then, just let me know. Are you ready to start?"
Begin first task	◇ Hand users the first task. ◇ Clarify the task if it's confusing. ◇ If necessary, prompt the users to begin interacting with the prototype.	"Okay, here's the first thing we'd like you to do. Take a minute to read this and let me know if it makes sense. If so, then whenever you're ready please show us what you would do first."

standing on the edge of the swimming pool, worrying that the water's going to be cold. The key to getting users to relax is having them jump in the pool and realize that the water's fine. Once they start interacting with the prototype, their brains become engaged in the task and they focus less on the social nature of the setting. It's also easier for the facilitator to reassure the users that they're providing value once they're doing something besides staring at the interface. So the key to starting a test off on the right foot is to get the users "clicking" and "typing" as soon as possible. (Exception: Sometimes it's interesting to start a task by asking users to describe how they currently do it or how they'd expect to go about it to determine whether your interface supports their approach. Just keep this discussion brief.)

It's amazing how quickly users get into the task and act in a realistic manner. In one test of a video conferencing system, the task involved negotiating a business deal. We used photographs of a person in various facial expressions to simulate the other end of the video connection, and users talked to the pictures as if they were live people. Even with a human obviously manipulating the prototype, users often say things like, "Hmm, the red light came on" or "Good, it worked," which indicate that they're focusing on the *interface* as the entity they're interacting with, not the humans simulating its behavior.

I also make it a point to talk to users afterward and ask what they thought of the experience. I've found that most people seem to understand and appreciate the reason why we're testing with paper. Occasionally someone will comment that some aspect of the interface might have been clearer on a screen. I'll simply acknowledge their statement (for example, "Yes, that table will be easier to read when the columns line up") and thank the user again for his or her help. But I've never heard a user say that the experience was silly or a waste of time, and I've heard plenty of positive feedback.

How the "Computer" Behaves

For the most part, the Computer should think of himself or herself as a machine that can do only what it's been programmed to do.

Accurately Reflect the Users' Inputs

The essence of any human-computer interface is that the machine takes input from the human, does some processing, and produces some kind of output. For an interface to be usable, users need to understand the cause-and-effect relationship between their inputs and the outputs, even if the processing is a complete mystery. (I don't need to understand how my car's antilock brakes work. All I need to know is that if I keep pressure on brake pedal—even when it makes that weird rattling noise—the car will stop as quickly as it can.)

Although developers often focus on the processing—which is, after all, the part that has to be coded—users deal mostly with the inputs and outputs, so it's important to have your prototype represent them accurately. For example, if the user writes "hiking boots" into a search engine field, use that piece of removable tape at the top of the search results page to make it say "Search results for: hiking boots." If you try to rely on memory, they (or you) may forget exactly what they entered. Similarly, if you show users a screen with different data than what they entered, they can get confused unless the differences are minor and easily explained.

Wait for the Users

Avoid anticipating what the users will do. To speed things along, sometimes an overly helpful Computer will set down the next screen (or more subtle, pick it up) before the users actually do anything to make it appear. Resist this temptation unless you're absolutely certain what screen they want and how they would have gotten there.

Avoid Conversation

Since, as the Computer, you're sitting across from the users and they know that *you* know how to use the interface, sometimes users will direct questions to you. Don't answer them—it's the facilitator's job to handle this. (If this feels awkward, you might say, "I'm sorry, but computers can't talk.") However, there are some situations when it's appropriate for the Computer to say something:

- **To distinguish action from discussion.** Sometimes users will point at the prototype while figuring out what to do next. In this case, the Computer (or the facilitator) can ask, "Are you just discussing or are you doing something?"
- **To clarify what the user would see.** If the prototype is messy enough that users can't read it, or there's something that's hard to represent with paper, it's okay to state what the user is looking at, *but not why.* For example, the Computer can say things such as, "That button is gray," or, "These are tabs." But beware the word *because,* which is a red flag that you've slipped into explanation. For example, "That button is gray because you haven't selected a record yet." If you catch yourself saying "because," stop talking.

- **To provide an oral error messages.** Sooner or later, a user will take a wrong turn that you aren't prepared for. It's fine if you need to make up an error message on the fly and simply tell the user what it would say. Try to use the same content that the real system would—resist the temptation to give them additional information that would not appear in the context of the error. (Also see the discussion of "incredibly intelligent help" in Chapter 4.)

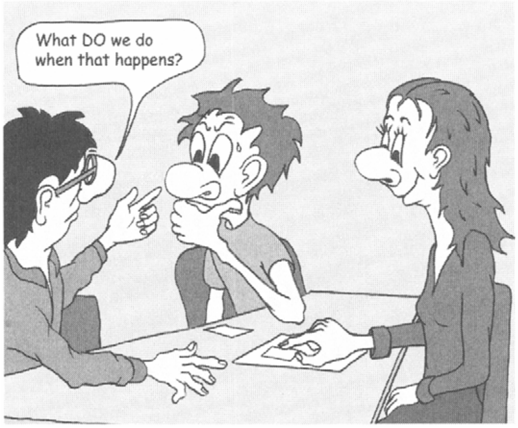
Facilitating a Paper Prototype Test

In a paper prototype test, the facilitator's responsibilities are essentially the same as in any other kind of usability test, but there are a few differences due to the medium of paper.

- **Have them show, not tell.** In the process of thinking aloud, users sometimes slip into verbal explanations, such as, "At this point I'd start over." (People do this when testing on a computer too, but it happens even more often in paper prototyping because the Computer is human.) Ask the users to demonstrate *how* they'd do that. Avoid letting the users simply talk through the task—remind them to show you. Sometimes it's okay to relax this a bit after you've established a common vocabulary. For example, if the users demonstrate that "start over" means "click Back until we get to the home page," you don't necessarily have to make them traverse all the interim pages each time if this method would work. But when in doubt, have them explicitly show you.
- **Clarify that it's okay to write on the prototype.** Sometimes users are reluctant to write on the prototype, and you may need to reiterate that you really do want them to write on the removable tape, transparency, or prototype itself. (An occasional user will go too far and put Xs on buttons to indicate clicking; gently remind them that writing only replaces typing, not clicking.) Whenever users ask if they can write something on the prototype besides data, I say yes because chances are it's something interesting, like crossing out information they don't want to see.
- **Watch for Computer mistakes.** Sometimes the Computer will make a mistake, either a simple one, such as not moving a highlight, or a major one, such as showing the wrong screen. The facilitator should try to keep tabs on the Computer's logic and note any problems. Sometimes a simple, "Is that right?" or "Didn't we change that?" is sufficient to clue the Computer in to the problem.
- **Handle valid (but unprepared-for) user actions.** Users may do something entirely reasonable that you simply haven't prepared for, such as typing a term

you didn't expect into the search engine. One tactic is to confirm that their way would have worked and then ask them to please find another. Or perhaps you can quickly sketch a screen or modify one of the screens you did prepare. For example, if you have a page of search results that contains more items than what the user's search would yield, maybe you can cross off items to approximate the list that the user would see.

- **Take a break if the paper prototype crashes.** Every once in a while, the users manage to end up in a state that didn't occur to the development team even in their wildest dreams (or nightmares). I can tell when this happens because the Computer will look at the other designers and ask, "What *do* we do when that happens?" This is valuable because it reveals a pitfall that the code will need to avoid. If this happens, take a short break so that the team can step out into the hall and confer. You may need to join them to hear how they'd like to proceed. Be sure to reassure the users that they've just done a very good thing in finding this problem—if the users are relaxed enough for humor to be appropriate, I'll tell them that the paper prototype has "crashed" and we need to go reboot it.



Paper prototypes can crash too!
(Illustration by Rene Rittiner.)

- **Answer questions when appropriate.** Given that paper prototypes are often used before the design is complete, sometimes the users raise really good questions that are outside the scope of what you're testing. If the issue is an important one and isn't going to be covered by our tasks, I'll simply answer it (or call upon the appropriate team member if I don't know the answer). For example, in a security application used by network administrators, one of them wondered whether the system would let them set the security level of the passwords created by end users ("Red Sox" is easily guessable, at least in Boston; "Ab84gh" is not). This was something we hadn't planned to cover, but it was an excellent question pertaining to functional requirements, so I let the product manager discuss it briefly with the users.

Ending the Test

Sometimes there isn't time to complete the last task of the session, especially if it's long or the users are having difficulty. In that case, I'll ask the Computer to demonstrate the remaining steps so that the users can get closure on what they were doing. As a bonus, we may still identify another problem or two if the users are surprised by something and comment on it.

I usually reserve the last 10 minutes of the test session for discussion (especially if observers are in the room) of any interesting issues that came up during the test. Although I might do this in any type of usability test, in paper prototype tests there is a greater likelihood that the observers will have questions for the users. For example, this is an appropriate time to ask users if the tasks were typical of their work or to follow up on their comments. Sometimes one of the designers has revised a troublesome screen and wants to ask if it's better. With the tasks behind us, it doesn't matter as much if someone asks a leading question, and it's okay to answer the users' questions directly. As a facilitator, I'll keep quiet if the observers and users are having a fruitful discussion.

I do keep my eye on the clock, however. It's important to let the users go when we said we would, even if the discussion is interesting. Usually I'll give a "2-minute warning" when we're running out of time, which allows for another question or two. If the users and observers are engaged in discussion, when we're out of time I will stand up as a cue that the session is over, and people get the hint. It's usually a good idea to escort users back to the lobby so that you can be reasonably sure that they've left, especially if you plan to hold a debriefing meeting immediately after the test. (It's an equally good idea to only say things about users that you wouldn't mind them overhearing, just in case a user returns to the test room to retrieve a forgotten umbrella.)

Combining Roles

In addition to users, there are three roles in a paper prototype usability test: the facilitator, the Computer, and the note-taking observers. If you have a very small team, you might need to combine roles. The best approach depends on the skill sets of the people involved—here are some thoughts.

- **Facilitator/observer.** This is probably the combination I'd try first. I believe that with practice, a facilitator can take good notes and still manage the rest of the facilitator duties, but other usability specialists feel that one or both activities may suffer. It's probably fair to say that it depends on the experience and comfort level of the facilitator.
- **Computer/facilitator.** It's difficult to test your own design. The Computer has in-depth knowledge of how the interface behaves, and often this knowledge came from designing it. Although it's possible to facilitate a test of something that you've designed, it's really hard to remain objective. Unless you're certain that you can resist the temptation to lapse into explanation/justification of the interface, it's best to let someone else facilitate. On the other hand, if you know how to facilitate but are unfamiliar with the interface, you can probably do a passable job as Computer after a couple of run-throughs, especially if a designer will be observing and can help if you get stuck. The advantage of this approach is that the designer can focus on note-taking.
- **Computer/observer.** This one isn't very practical—the Computer has his or her hands full (literally) with the prototype, so it's hard to take more than a few hastily scrawled notes.
- **Computer/facilitator/observer.** Not recommended—I did this once, and my brain almost exploded.

No matter which way you handle it, the session may go a little slower than usual if you're juggling more duties than you're used to, but it still beats not testing at all. Reserve some time after the test to jot down additional notes. You may also want to consider videotaping.

Iterative Refinement: Modifying the Prototype

With paper prototyping, it's okay to make changes as soon as you have evidence that there's a problem. It's not necessary to keep the prototype and tasks the same for each test because you're not trying to gather statistics—you're trying to identify and fix problems.

So what happens when you find a problem with your paper prototype? If it's a little problem, fix it right then and there. If it's a bigger problem (or you're not sure whether it really is a problem), wait until after the test.

Making Changes during a Test

Making changes during a usability test always reminds me of *Harold and the Purple Crayon,* one of my favorite books when I was a child (Figure 9.3). Harold is a plucky little guy who goes through his cartoon world by drawing all the adventures that he wants to have happen, such as going sailing. When his adventures get a bit too much, he grabs his trusty purple crayon and draws his way out of danger. So when a user says, "I want to look at the return policy, but I don't see a way to do that," you can grab your equivalent of a purple crayon and write "Return policy" on the screen so that the user can click it. Similarly, it's fairly simple to change wording, rearrange items (with your purple scissors), add or remove a control, and so on. You'll get immediate feedback on whether the change helps, and if so it's already in place for the next test.

After he had sailed long enough, Harold
made land without much trouble.

Figure 9.3 Like Harold with his purple crayon, sometimes you can change your interface on the fly to support what users are trying to do. (This page is my favorite because of the pun on "made land.")

Caveat: Users Aren't Designers

As a rule, it's best not to give users the virtual purple crayon and let them redesign screens in a usability test. Although users are good at describing problems, most are not good designers. My favorite illustration of this comes from an episode of *The Simpsons* in which Homer, egged on by a rabid marketer, is given free rein to design his dream car (Figure 9.4). Naturally, when he finished adding features the car was a rolling fiasco.

> *Homer:* I want a horn here, here, and here. You can never find a horn when you're mad. And they should all play 'La Cucaracha.'
>
> *Advisor:* What about a separate soundproof bubble-dome for the kids with optional restraints and muzzles?
>
> *Homer:* Bullseye!

With that caveat, sometimes it is useful to ask users to mark features or content they're most interested in or to cross off things they don't want. One Web site I tested used six tabs at the top for each of its major content areas. At the end of the usability test, I gave users a pen and told them they had 10 votes to distribute across those six tabs to indicate the value they perceived in each. Although this experiment was crude, there were some useful patterns, like the fact that one of the tabs contained stuff that none of the users cared about. If you are interested

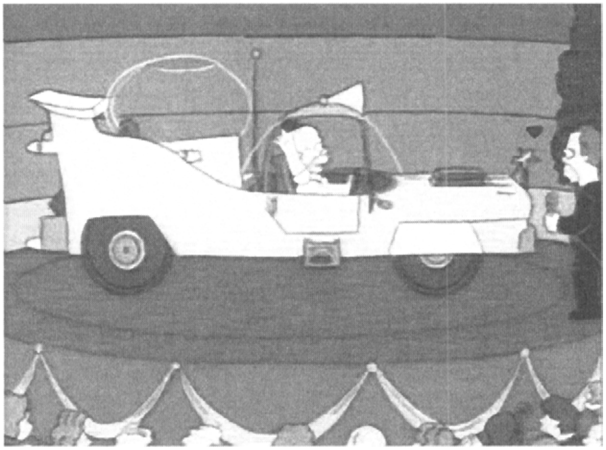


Figure 9.4 Homer Simpson knew exactly what functionality he wanted in a car, but he was still a lousy designer. (© Fox Twentieth Television.)

in more information about ways to involve users in the design process, you'll want to read other books and papers about participatory design (see the Reference section).

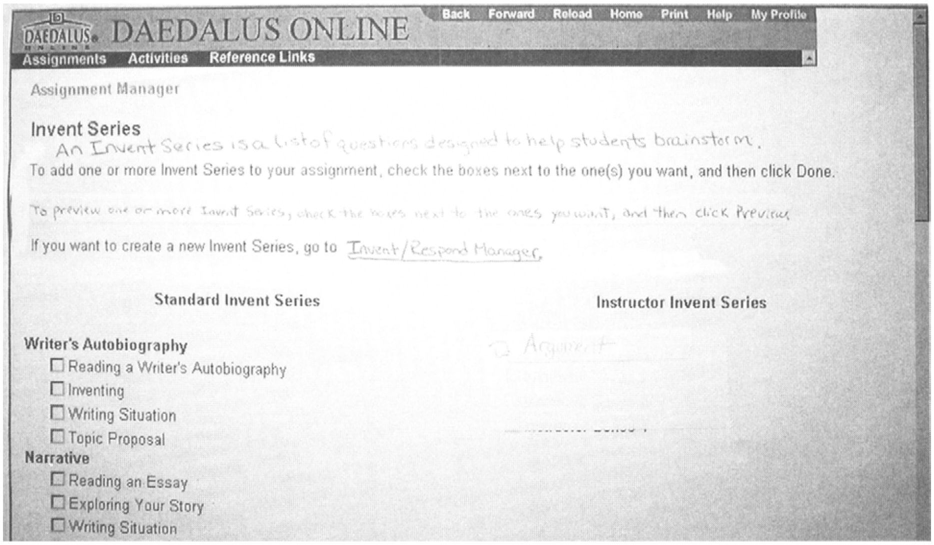
Making Changes between Tests

Hopefully, you'll be conducting a handful of tests with your paper prototype. You should allow yourself some time between tests—say 2 or 3 hours—to make more substantial changes than what you were able to do on the fly. (If the design is still in its infancy or many changes are expected, you might even designate a day in the middle of your usability study for rethinking the design.) Rearranging screens, simplifying a screen, adding an example—these kinds of changes often help considerably and aren't difficult to do. Even when more substantial redesign is called for, it often doesn't take as long as the initial design did because your brain has a better grasp of the problem.

◇ **Problems first, then answers.** One method that works well for making changes efficiently is to first list the issues on a whiteboard—everything that the observers saw during the test that indicated a problem. (If you do this in the same room that you're testing in, remember to erase this list before the next users arrive!) Then divide and conquer, just as you did to create the prototype—people put their initials by the things they want to solve. With this approach it's possible for a team to make fairly substantial changes to the paper prototype in an hour or so. If you're doing this over lunch, order a couple of pizzas for the team and then you'll have plenty of help.

◇ **Include information in the interface.** The interface is the first and often the best place to explain things to the user. Always ask yourself whether a particular usability problem can be solved by directly changing the interface rather than writing instructions somewhere else. Try to make the easiest change first, which often means wording. For example, in testing a Web application for teachers, we found that users were confused by what to do next. We solved this problem by adding a sentence to the bottom of the page suggesting the next step. This worked so well that we called these "magic sentences." (See Figure 9.5.)

Note: These users were English teachers, and their willingness to read instructions online was perhaps a bit greater than usual. But I have seen equally good results from changes of similar scope—adding an example next to a field or reformatting a page to put the most-needed information at the top.



Assignment Manager

Invent Series

An Invent Series is a list of questions designed to help students brainstorm.

To add one or more Invent Series to your assignment, check the boxes next to the one(s) you want, and then click Done.

To preview one or more Invent Series, check the boxes next to the ones you want, and then click Preview

If you want to create a new Invent Series, go to Invent/Respond Manager.

Standard Invent Series

Writer's Autobiography
- □ Reading a Writer's Autobiography
- □ Inventing
- □ Writing Situation
- □ Topic Proposal

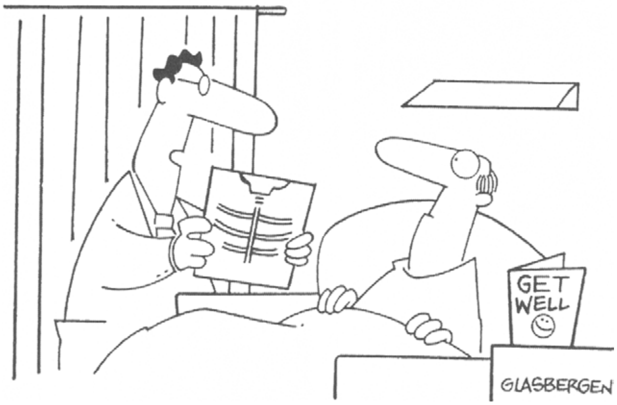
Narrative
- □ Reading an Essay
- □ Exploring Your Story
- □ Writing Situation

Instructor Invent Series

□ Argument

Figure 9.5 The hand-written part is what we called a "magic sentence." These brief explanations helped users understand the concepts and/or next step.

© 2000 Randy Glasbergen www.glasbergen.com



"Your x-ray showed a broken rib, but we fixed it with Photoshop."

Figure 9.6 Make sure you're not just tweaking your interface to create the illusion that you've solved a particular problem.

- **Don't suboptimize the tasks.** Although your paper prototype is built around a particular set of tasks you've created, don't forget about all the other tasks that users might be doing in real life. Sometimes it's tempting to suboptimize the interface, in other words, tweak it so that users will get through a specific task better. For example, users aren't seeing a menu option that falls below the fold of the site, so you move it up. But to make a change like this, you also have to decide what gets pushed *below* the fold. Take a step back and be sure this makes sense for the interface as a whole. Otherwise you're only creating an illusion that you've solved the problem (as the cartoon in Figure 9.6 illustrates), and the illusion will shatter as soon as the user does a different task.

This chapter has discussed what the facilitator and Computer should do in a paper prototype usability test. The ways that you set up the room, greet users, introduce the concept of paper prototyping, and facilitate the session are all important. But don't forget that the main purpose of usability testing is to provide data to other members of the team so that they can make the product better. The next chapter focuses on the observers.