Chapter 5

Planning a Usability Study with a Paper Prototype

So you've decided to give paper prototyping a try. How do you go about doing it? This chapter outlines the people, activities, and time you'll need. Some companies do their first paper prototype usability study by hiring a consultant to come in and lead all the activities. This chapter and the others in this section cover much of what I do as a usability consultant—theoretically, it should put me out of business!

I'm not being facetious. I believe that the best way to make interfaces more user-friendly is to close the feedback loop between product teams and users by teaching people how use paper prototypes to communicate, ask questions, and find answers. Once everyone knows how to do this, I'll find some other way to earn a living!

If you're not a usability specialist, you may be doing some of these things for the first time. If you already have a process for conducting usability studies, this chapter will help you identify the activities that are different from what you typically do. (On the surface, the only thing that's different about using a paper prototype for usability testing is that you have to create the paper prototype. But there may be some other differences from your typical process, such as leaving time for changes between tests and less need for formal reporting.)

The process described in this chapter has worked well for me on dozens of projects, but it certainly isn't the only way to do things. There are other books that go into detail about the activities involved in usability testing; the book by Dumas and Redish (1999) is a good one to start with, and you'll find others in the References section.

Overview of a Usability Study Using a Paper Prototype

Table 5.1 shows the "view from 30,000 feet" of a usability study using a paper prototype. (As a reminder of the conventions I'm using, a usability *study* is a series of tests conducted with a given interface and set of tasks. A usability *test* is just one session with a user or user pair.) There's more detail about each activity later on. Reduced to its essentials, the process looks like this:

- Determine the goals of the testing—what do you want to learn?
- ❖ Who are the users? Define them so recruitment can begin.
- Create tasks around things that those users do.
- ❖ Create the paper prototype pieces needed to perform those tasks.
- Hold internal walkthroughs to prepare for testing.
- Conduct several usability tests, refining the prototype after each test.
- Establish priorities for the issues found.
- Plan additional changes to the interface in the short term and/or track them so that they can be resolved later.
- ❖ Communicate your findings to others who weren't directly involved.

Notice that you don't start the process by creating the paper prototype. Rather, you first agree on high-level goals and concerns, then develop usability tasks around those things, *then* develop the paper prototype and test it. How early can you start? If you're sketching screens on a whiteboard, you're far enough along to start paper prototyping. If you have questions about what users will like, understand, or accept, you can throw together a prototype and get answers.

Table 5.1 Overview of a Paper Prototyping Project

	What Happens	Who	Time
Kickoff meeting	 Discuss goals, risks, and concerns 	All stakeholders	3 hours
	Agree on user profile		
	Oetermine "core team"		
	♦ Set schedule		

Table 5.1 Overview of a Paper Prototyping Project—cont'd

	What Happens	Who	Time
User recruitment	❖ Find people who match the user profile and schedule them	1–2 people if you're doing this in-house; some companies outsource this	Depends; a 2- to 3-week lead time is typical
Task design	♦ Create the tasks to be used in usability testing	Core team plus anyone with important input about what gets tested	3–5 hours
Prototype creation and walkthroughs	 ❖ List interface elements that are needed to support the tasks ❖ Split up the work ❖ Hold periodic walkthroughs ❖ Formal run-through without real users prior to usability testing 	Core team; others can come and go as sched- ules allow	$\frac{1}{2}$ to 5 days total
Usability testing and iterative refinement	 ❖ Perform usability tests (most last 1-2 hours) ❖ List issues after each test ❖ Revise the prototype before the next test 	All stakeholders	2 days total
Prioritizing issues and action plan	 ❖ Prioritize unresolved issues ❖ Discuss top issues and possible solutions ❖ Create action plan to address issues ❖ Track issues 	Everyone who attended one or more usability tests can help prioritize; the rest is up to the devel- opment manager and/or team	3 hours

Table 5.1 Overview of a Paper Prototyping Project—cont'd

	What Happens	Who	Time
Communication of results	As needed: Write summary of "top 10" issues Post results on intranet Write report Give presentation of results Create a walkthrough video Create an interface spec	1–2 team members (often, but not neces- sarily, the usability specialists)	1 hour to several days depending on the activities



www.paperprototyping.com has a document containing Table 5.1 that you can use as a blueprint for your own plan. Modify it as needed, and hand it out at your kickoff meeting.

People to Involve

Paper prototyping and usability testing should involve all members of the product team, not just those who have "designer" or "usability" in their title. Every day of a project, dozens if not hundreds of decisions are made that affect some aspect of the user experience. Even under-the-surface technical factors such as database design can have an impact on the user interface (and vice versa). Practically speaking, the number of decisions that affect the user is too large for one person or even one department to handle. If only one or two people in the company are responsible for the entire user experience, their limited ability to collect and disseminate usability data will quickly become a bottleneck. Instead, it's usually better to have many people participate in usability activities and get the data first-hand. This section explains who should be involved and to what extent.

Terminology: Designer and Developer

As I start tossing around words such as *designer* and *developer*, I'm going to run afoul of differences in how people define these terms. At some companies, the same person who designs the interface also codes it, tests it, and maybe even helps document and support it. Other companies distinguish between designers and developers. Furthermore, there are graphic designers and interaction designers and software designers and instructional designers—see the problem? I'm going to adopt some role-based conventions:

- ♦ A *designer* figures out what it should do or be. (The "it" depends on the context—usually I'm talking about an interface, but these terms also work for documentation, training, and so on).
- ♦ A developer makes it happen.
- ♦ *Technical* means "understands the constraints of the technology and tools used to implement the interface."

Using these conventions, someone from Marketing might be considered a designer, and a trainer wears the developer hat in the context of creating a course. People from a variety of departments might be considered technical depending on what they know—for example, some writers are very savvy about technology. I am deliberately using broader definitions than many people are used to because it underscores my philosophy that creating a product is a multidisciplinary process. You know who you are—if it sounds like I'm talking about you, assume that I am, even if I'm not using the title you're accustomed to.

The "Core Team"

If you have a large development team, it's a good idea to designate two to five people as the "core team" for the activities in a usability study. When it comes to scheduling, it's unlikely—and probably not even desirable—that the entire development team can be involved in every aspect of the prototype preparation and testing. Plan the activities around the availability of the core team and invite others to participate in whatever they're able to.

The core team consists of those people whose involvement is crucial to preparing and testing the prototype. Typically, the core team includes the designer/developer(s) responsible for the interface and a usability specialist if the team has one. One of these people takes the leadership role in planning and conducting the

activities. The development manager or product manager may be part of the core team, although if this manager oversees several projects he or she may delegate the responsibility to one of the people mentioned earlier. It's also quite common to have a tech writer, marketer, and/or graphic designer on the core team.

Technical Expertise

For a moment, I'm going to divide the world into two types of people:* those who are technical as I've defined it and those who are not:

- If you're the technical type, realize that other people have good insights about what users want and need, and you really want to get those insights before development is in full swing. The ideas that come from nontechnical people aren't always useful or workable, but the paper prototype will sort the wheat from the chaff. Be open to prototyping something even if you're not sure how to make it work—maybe you'll be able to use the idea in another form if it works well for users.
- If you're not technical, remember that interface design is a skill as well as an art. You have a lot to contribute, but not all of your ideas will be practical within the constraints of your development process. Sometimes you'll have to defer to the techie types when they say, "The architecture doesn't support it" or "That goes against our style guide." But don't be afraid to pick up a pen and scribble a screen if inspiration strikes.

It's important to have at least one person on the core team who has the technical perspective—knows the system architecture, the limits of the technology, what is easy, and what is hard. This will prevent the team from developing a prototype that is impossible to implement. This person(s) should have the final word on whether and how something is included in the prototype.

The Rest of the Team

Paper prototyping is a multidisciplinary activity and should include people in addition to those who are directly responsible for the interface design/development. In particular, seek out those who have direct contact with users: sales, mar-

^{*}There are two types of people in the world: those who believe you can divide the world into two types of people and those who don't.

keting, tech support, trainers. These people often have valuable insights about what users want and what confuses them. So do writers and QA/QE/test engineers because they are often among the first people to use the interface, albeit in a lab setting.

A Note about the Graphic Designer

If there's a graphic/visual designer assigned to the project, it's great to have that person involved, possibly as part of the core team. But the rest of the team shouldn't be intimidated by the presence of someone whose doodles could be framed and sold as art. When a team member has artistic ability, others may be tempted to let that person do all the prototyping work "because it will look nicer." Remind everyone that it's fine if the prototype looks like it was drawn by a 10-year-old because the point is *not* to spend time making things look pretty when you'll want to change them tomorrow anyway. The graphic designer shouldn't have to do more of the work than anyone else.

Kickoff Meeting

When starting a paper prototyping project, get everyone together to discuss the goals of the testing and the method. Here's a typical agenda for this meeting:

- Provide an overview of activities
- Discuss risks and concerns
- ♦ Create a user profile
- Operation Description

 Determine the schedule

 Determine the sc

Provide an Overview of Activities

The first time you do paper prototyping or even usability testing, it's natural for people to have questions about these activities. You might print out a version of Table 5.1 and take it to the meeting. You might even make a paper prototype of an application everyone is familiar with, just to show them what you're talking about. Paper prototyping is one of those visceral things that can be hard to explain, but everyone knows it when they see it in action.

People who are not part of the core team may wonder which activities they need to be involved in. I recommend that they attend the following:

- ❖ At least one walkthrough, preferably the last one before testing begins. People get more out of observing tests if they're familiar with the prototype beforehand.
- ❖ At least two usability tests. Mathematically speaking, you can extrapolate in any direction from a single data point. If someone observes only one usability test, he or she may prematurely conclude that the interface has a problem and that other users will have the same difficulty (or conversely, that this user represents an outlier and that there really isn't a problem with the interface). But if that observer watches another usability test, he or she will see similarities and differences, which provides a better perspective on reality. Even if some people can observe only one test, that's still much better than nothing—just caution them that what they see may not be typical and that they should compare their observations with what happened in the other tests.
- ❖ The debriefing meeting. Because this is when the issues are prioritized and discussed, it's a good way to find out what happened at the other tests. If there will be a report of the findings, you might suggest reading the report as an alternative to attending the debriefing meeting, although the downside is that the person won't be involved in prioritizing the issues.

Discuss Risks and Concerns (Finding the Rattlesnakes)

As the saying goes, it's not the rattlesnake you see that will bite you. The real value of paper prototyping comes from its ability to point out problems early in the development process, while it's still easy to avoid them. Thus, you should design your usability tests around whatever aspects of the user experience you are most worried about (or know the least about). This is called *risk management*.

This kind of thinking is backward from how software is often developed. To have a solid base of code to build on, developers often start with the pieces that are well understood, just to get something running. Aspects of the design that are still up in the air are postponed until later, thus providing ample cover for the rattle-snakes. But because paper prototypes don't depend on functioning code, they allow you to do the opposite—start with whatever you have the most questions about, mock something up, and get user feedback. As soon as you have a handle on a particular issue, you can turn your attention to whatever's next on your list of things to worry about.

Paper prototyping is an excellent tool for risk management because it helps you clarify what you do and don't know about how well your interface will work, and it can help the team make important decisions. Ronnie Thomson, Director of Engineering, describes Centra's prototyping experience before the first release of their Web application Symposium: "There was an undercurrent of doubt about whether we were doing the right thing [with the 3D interface]. Paper prototyping helped the company address these issues and change technical direction. Without it, these decisions would likely have taken 6 to 12 months longer."

At the kickoff meeting, ask everyone to discuss what things they're concerned about and/or would like to know more about. The exact questions, of course, depend on the circumstances of your project, but following are some questions you might use to get people started:

- ♦ What are you least sure about?
- ♦ What are the greatest risks from a business perspective?
- ♦ What problems keep you up at night?
- ♦ What important design decisions do you still have to make?
- What have we gotten the most negative feedback about?
- ♦ What tasks are critically important to users, even if they're done infrequently?
- ♦ What parts are new?

If you have an existing version of the interface, make sure to get input from trainers and/or customer support, who will likely have a good idea where the trouble spots are. If your team is designing something new, these questions may be harder to answer. You may not even know where your biggest risks lie—in essence, this is your biggest risk.

Write down everyone's concerns on the whiteboard, and ask someone to type it up at the same time. It's fine if people don't agree on what the problems are. Don't discuss, just list. The next chapter on task design explains how you use these concerns as inputs into the process of creating tasks. (Although you could wait until the task design meeting to have this discussion about risks, I find it's useful to do it at the kickoff meeting because it helps people understand the purpose of paper prototyping and usability testing.)

Discussing your concerns first may help clarify the type of user you want to focus on, which is the next activity in the kickoff meeting. However, it's possible that these steps should be reversed for your project—the things you're most concerned about might depend on which user(s) you're talking about. If you find yourselves getting bogged down in the discussion of risks, you may want to switch gears and talk about users, then pick up your discussion of risks once you've agreed on the user profile.

Create a User Profile

It's important to get consensus from the development team about which kind of users you want to recruit for the usability tests. It's common for interfaces to have more than one type of user, but if you're doing usability testing for the first time, choose only one. For example, if your interface is used by system administrators and end users, pick one of those populations to test with. The main problem with testing multiple types of users is that you're tempted to subdivide your test population too much, and it's very hard to interpret the results if you've brought in only two users from each of four target markets. You need to test with enough of one kind of user to see the patterns, and once you've done that you'll have plenty of data to digest. So pick one type of user for now—you can always do another usability study later.

There is no one best process for creating a user profile. I describe what I usually do, but you might want to do some further reading on the topic.

Start by discussing what characteristics the target users have. Following are some questions you may find helpful:

- What education or training have they received?
- ♦ What are they responsible for in their work?
- How familiar are they with the technology and concepts our product uses?
- ♦ What products similar to ours have they bought/used?

Make a list of all the characteristics mentioned. Once you've run out of steam, look at your list and identify a subset of four to six characteristics that capture the essence of this user. (Some examples are provided shortly if you'd like to peek at them now.) Typically, you'll find that some characteristics supersede others. For example, you may not care if a network administrator has a computer science degree as long as he or she has somehow gotten a job that includes responsibility for configuring servers.

Once you have agreed on a user profile, ask yourself if the people you'd like to bring in for usability testing might be slightly different than this profile. Sometimes you may be able to broaden a criterion; other times there may be certain people you want to screen out.

Example of broadening: "We'd really like to get network administrators who do the hands-on configuration and troubleshooting of the network. But it's okay if we have a couple of their managers too, because they still understand the concepts even if they don't do the hands-on work."

Example of narrowing: "We should screen out people who design or develop Web sites—they know too much so they aren't typical of the audience." (Another obvious example is people who work for your competitors. If an outside company is doing recruitment, this won't be obvious to them, so you'll need to be explicit about what competitors you're excluding.)

Examples of User Profiles

Following are some sample user profiles that are similar to ones I've used for real projects.

For a Data Security Web Application

- ♦ Works at a large organization (>1000 people) such as a hospital or Fortune 1000 company. (The size of the company isn't important per se, but it acts as a proxy for a company that has a need to protect internal data. If someone works at a smaller company but meets the other criteria, he or she is in.)
- Manages a nontechnical group (such as Purchasing, HR, Finance, Marketing, etc.). Screen out Engineering, IT, MIS.
- Approves access to sensitive resources (such as network dial-in, high-value transactions, financial accounts, employee records, medical records, and proprietary data).
- Has budgetary responsibility; approves expenditures.

For a Music Web Application

- ❖ Buys at least two music CDs per month (more is better).
- Has downloaded music from the Internet.
- ❖ Is age 18 to 34.
- ♦ Owns (or has used) a portable MP3 player: [list]
- ♦ Listens to at least one of the following radio stations: [list]

For Small Business Banking Customers

- ♦ Works at a small business with annual revenues between \$250,000 and \$500,000 per year.
- Does not work in any of the following industries: banking, financial services, or insurance.
- ♦ In their company, is the decision maker or influencer for the company's banking, payroll, insurance, and/or retirement plan decisions. (Ideally, we'd like people who are involved in all these areas.)
- ♦ Is not a software or Web site developer.
- ♦ Company currently has a business account with [bank].

Notice what these profiles have in common—they consist of no more than a handful of factors, they distinguish between requirements and nice-to-haves, and they also mention characteristics to screen out.

Demographics versus Characteristics

In practice, I've found that demographic factors such as age, gender, and income don't seem to have much effect on a user's ability to turn up usability problems, so I often don't include them in the user profile. Usually, if you get people who have the appropriate skills, knowledge, and motivation, demographic factors are secondary.

However, I'm not saying that demographic factors are irrelevant—they are certainly an important part of understanding who your users are. It's just that when it comes to finding people to participate in usability tests, demographic factors often turn out to be only a proxy for what you're really interested in—usually some combination of knowledge, preferences, and behavior.

For example, say you're developing a travel Web site, and marketing research has shown that your target market falls into a certain income range. But then you think of that low-income couple you know who treat themselves to a week in Aruba every spring. Perhaps the important factor isn't income, but the number of trips taken each year, in other words, their behavior. So you'd ask, "If the person buys more than one plane ticket online per year, would we want them regardless of income?" If the answer is yes, you can drop the income demographic from your user profile because it's only acting as a proxy for amount of leisure travel.

Another way to scrutinize demographic factors is to ask, "How would someone with factor X behave differently when using our interface than someone without it?" If you can't answer that question, the factor can be omitted from your profile. For example, several years ago I worked with a company that was introducing high-speed Internet access to a market. They wanted to conduct a paper prototype test of their Web site. One of the factors on their initial user profile was "Has two phone lines." Their reasoning was that people who had a second phone line for their computer would be prime candidates for another method of Internet access. This made perfect sense for their *marketing* efforts, but when we asked ourselves, "How would people with two phone lines use this Web site differently than people who only have one?" we realized it was totally irrelevant for what they wanted to test. Which was good news because dropping that requirement made user recruitment much easier.

Although I often omit demographic factors from a user profile, I'll make an exception if a factor truly is an integral part of what we're trying to study; I have tested Web sites specifically aimed at men, women, college students, parents, and so on. In these cases, the product team believed that a demographic factor had a strong effect on the validity of the data obtained from testing. (Translation: If we had tested with someone else, there's a significant risk that the usability test results would have been dismissed as atypical.)

Reality Check

Just because your team reaches consensus on the user profile doesn't mean that it represents reality. (In the more egregious cases, you get a big clue during user recruitment when you can't find those people—because they don't exist!) Take every opportunity to validate your user profile with people who know (or even better, are) users. One of my colleagues once worked at a company where the product team assumed that users would have a background in math and statistics. But when my colleague visited some training classes, he found that only about one third of the users fit what the product team had considered to be the "typical" profile. Fortunately, he discovered this in time to fix the product, which might otherwise have alienated two thirds of its target market.

Time Needed to Create the User Profile

The first time you create a user profile, it might take you an hour or three, especially if there are differing assumptions about who your users are. Sometimes you may realize that you don't have a clear picture of who your users are—this isn't ideal, but it happens. If there are uncertainties or disagreements about users, note these as areas for research.

Once you have a user profile, you might be able to reuse it for subsequent usability tests. But be sure to first discuss whether you've learned anything about users since the last usability study that might cause you to modify the user profile.

Determine the Schedule

A typical first-time usability study using a paper prototype takes a total of 5 or 6 days of effort, but spread across perhaps a month from start to finish. To recruit users, you have to know when the tests will be. Finding users has a longer lead time than preparing the prototype, so start by scheduling the usability tests and allow 3 to 4 weeks for user recruitment. (Once you have a user profile and a recruitment method established, it may be possible to test on much shorter notice.)

Like they said on Star Trek, "Space: the final frontier." Your testing schedule might depend on when you can get a place to do it. Try to reserve the same lab/conference room all day on the days you're planning to usability test; although paper prototypes are portable, they can be a pain to move.

Work backward from the first test to schedule the activities you'll need to get ready. Although you may worry that you won't be prepared, chances are you will be—in dozens of paper prototyping projects, I've never known a team that couldn't get ready in time. One of the advantages of paper is that you can throw something together at the last minute if need be.

Length of Usability Tests

Perhaps 95% of the usability tests I've done have been either 1 or 2 hours in length. For your first couple of usability studies, I suggest that you choose between 1-hour or 2-hour tests. Go with 2 hours if you have many things to cover, if the interface contains new or complex concepts, or if it supports a work flow that contains more than a handful of steps. For example, in testing software for network administrators, I've always planned 2-hour tests. On the other hand, 1 hour is usually sufficient if the interface is smaller in scope.

That's rather vague advice, but fortunately I've found that the product team usually has a pretty good gut feeling about whether their interface needs 1 hour or 2. Ask yourselves, "Can a user cover what we're interested in within 1 hour?" and trust your collective instincts. If you're evenly divided, go with 90 minutes.

Of course, tests of other lengths are possible. In my experience, tests longer than 2 hours are rarely needed—if you plan one, be sure to include time for a break. Tests much shorter than an hour may not be not practical because you'll want time at the beginning for introductions, time at the end for Q&A, and you can

probably come up with enough tasks to keep the user busy for a while. Short tests might make sense if you're redesigning only a small part of the interface or you have just one or two tasks; this is more likely to be the case if you're already conducting frequent usability tests.

Number of Users

As a rule, testing with five to eight users will provide enough data for you to see the main patterns, provided that you use roughly the same set of tasks (it's okay to vary them a little) and the users are from the same profile. There is some debate on this number-of-users topic among usability professionals, and both sides are able to back up their claims with some solid evidence. Nielsen and Landauer (1993) put forth their finding that the curve describing the total number of problems found started to flatten out after about five or six users (Figure 5.1). More recently, User Interface Engineering (Spool & Schroeder, 2001), reported that when users are asked to bring their own tasks (and credit cards) to e-commerce sites, it takes considerably more users to find the major problems—on the order of dozens. This

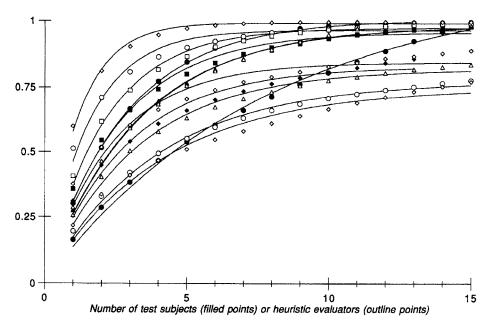


Figure 5.1 This graph from Nielsen and Landauer's work (1993) is often cited as the reason why you only need a handful of users—there's clearly a point of diminishing returns in testing with additional users.

is because when users do different things, they cover different aspects of the interface. Not only that, but when users are pursuing goals they truly care about (not to mention spending their own money), they are more sensitive to subtle issues than when they're doing a task someone else has invented.

Fortunately, with a paper prototype, the tasks are naturally somewhat constrained because you're only prototyping a portion of the interface and/or you're preparing the data the user sees. Thus, paper prototype tests tend to fall under the conditions of the Nielsen and Landauer study. Practically speaking, I find that after three or four tests I can usually see the patterns in what users are confused about and bothered by. There is no guarantee that you'll uncover all the problems, but once you have a few hours' worth of observations, it's time to step back and digest what you've learned.

So I suggest that for your first usability study, you plan no more than four to six tests. If you are using co-discovery (two users at a time, as explained in Chapter 8), you might schedule four 2-hour tests (for a total of eight users), or if you would rather test with one user a time, you could plan six 1-hour tests. With either of these schedules you can complete the testing in 2 days. Keep in mind that you can always do more tests later if you want.

Spacing of Tests

It's important to leave time between tests to review what you've learned and make changes to the prototype. If you're accustomed to conducting nonpaper usability tests, you may be in the practice of scheduling an entire day's worth of tests back-to-back, but this doesn't work as well with paper. I suggest a maximum of 4 hours of testing time per day, with a minimum of 2 hours between tests.

The schedule I've most often used is to hold usability tests from 10 AM to noon and from 2 to 4 PM on two consecutive days. That's a total of four tests, and with co-discovery that's eight users. Of course, testing can be spread over a longer period if the team expects to make substantial changes. In that case, I might leave a day in between the 2 days of testing. I rarely allow more time than that because usability testing with a paper prototype is an immersive activity. If you stop for too long, you'll lose momentum.

Estimating the Other Activities

Once you have scheduled the usability tests, you'll want to schedule the activities that precede and follow them.

Task Design

The next chapter discusses task design in detail. I've found that task design is often the most difficult activity in a paper prototype usability study because it tends to expose missing or conflicting assumptions that team members have about the product and what users will do with it. Thus, from a planning perspective I like to set aside a minimum of 3 hours (per user profile) to design tasks—and be prepared for it to take up to twice that long, especially the first couple times. It's also a good idea to have the product team review your tasks; thus, you may want to plan some time to modify them.

Creating the Paper Prototype

Creating the paper prototype takes anywhere from half a day to a week, depending on the state of the design. Start by assuming 2 days and adjust the estimate depending on various factors—less time if you're working mostly from screen shots and up to a week if it's a complete redesign or many people have input. If you think it will take longer than a week, chances are you're overpreparing. Also keep in mind that you don't need to prototype the entire interface, just enough to do the tasks.

Allocate time in half-day increments. Paper prototyping is not an activity that can easily be done in a few minutes between meetings. Because of all the pieces, it takes a while to set up the prototype and put it away. (Some teams don't like to leave their prototype spread out on the table overnight because they're afraid that the cleaning crew will throw it away!) Most paper prototypes I've worked on have been developed in three to four afternoons over a week or two. You may also want to reserve a conference room for the prototype creation sessions and walkthroughs—you want a large surface to spread out all the pieces.

Walkthroughs

Chapter 7 explains the purpose and method of conducting internal walkthroughs prior to the first usability test. Walkthroughs are an integral part of the prototype creation process, so by setting aside time for prototype creation you've also accounted for the walkthroughs. The exception is the last walkthrough before testing-you may want to schedule that because it often involves a larger group than the core team that created the prototype.

Debriefing Meeting and Action Plans

You may want to plan a short session immediately after each test to list what you saw and then make changes to the prototype. Because that session is attended by the people who observed the tests, one simple solution is to tack on a half hour to the testing time—if the tests will last 2 hours, tell people to set aside 2.5 hours for each test they attend.

At the end of the usability study, after all the tests are complete, plan a debriefing meeting with the entire product team to prioritize and discuss what you learned. People's memories fade pretty quickly (my memory has a half-life of about 24 hours), so it's best to hold the debriefing as soon as possible after the last test—I try to schedule it for the next day. It usually takes about 2 hours to prioritize the issues, sometimes less, at which point you might continue into a discussion of how to address them, or you might schedule a separate meeting for that. Once the debriefing meeting is over, there is probably some additional work needed to document and/or communicate the results. Chapter 11 covers these topics in more detail.

At this point, I return you to your regularly scheduled development process—whatever methods you normally use to prioritize and plan your work, you should input the test results into.

User Recruitment

There are many good references on user recruitment, so here I only to touch on the main factors that affect planning and scheduling. First you must decide whether to manage user recruitment internally or outsource it to a company (such as a market research firm) that specializes in doing this sort of thing. The best decision depends on the following:

- Time. It's time-consuming to play phone tag with people, screen them, schedule them, send out paperwork and reminders, and so on. Some of this work can be done through email, but it still takes a nontrivial amount of time to manage the process.
- 2. **Money.** Depending on the user profile and your geographic area, you can pay upward of \$100 per user to have an outside company do user recruitment (not

counting the money you have to pay the users). This may be outside your means, or it may be money well spent.

- 3. Frequency of usability studies. One R&D department I know of has a contractor who works 20 hours a week scheduling usability tests. This makes sense for them because the department has several projects underway at a time, and they do frequent usability studies. Consider whether others at your company might also want to conduct usability studies; perhaps you can share recruitment resources.
- 4. Sources of users. Sometimes you may have no clue where to find users, or you may want people who are unfamiliar with your product or company. (If the latter, you obviously can't put an invitation on your own Web site.) Brainstorm about how your users get their news and professional information; to reach them, you might use newspaper or radio ads or flyers posted at a local college, professional organizations, user groups, and so on. If you're still struggling for ideas on how to find users, working with a recruiting firm may be your best bet.
- 5. Whether your users are customers. If you know who your customers are, it may be easiest to just contact them yourself. (A market research firm can work from a list that you provide and save you some time, but this usually doesn't save you money because they have to do almost as much work.) Be sensitive to the politics involved in dealing with customers—if people in your Sales/Marketing department have existing relationships with customers, you should discuss your plans with them first to determine the best way to proceed.

If you work with an outside company, they will likely want you to write a screener, which in essence means turning your bullet-list user profile into a series of questions, like a flow chart, that lets the company know whether a particular person meets your needs or not. The company you're working with should be able to provide you an example of the format they'd like you to use. Creating a screener can take a couple of hours, but it can often be reused for subsequent usability tests as long as the user profile remains the same. A screener can also be useful if you are doing recruitment internally, although if a member of your core team is doing recruitment they may not need one.

If you decide to do user recruitment internally, think about who might be well suited for the task. If you have an extroverted co-worker who's good at administrative details, managing user recruitment may be a perfect career enhancement opportunity for that person, as well as something he or she would enjoy.

How Many Usability Studies?

So far, this chapter has assumed you're doing one usability study. But how many usability studies do you need to do for your product? For starters, plan just one study and let the results determine whether you're comfortable that your design works or you need to redesign and then test again. Some product teams use paper prototypes in the early stages to smoke out the show-stoppers and then plan a small usability study or two later with the real interface to find any issues that wouldn't have come up with paper.

Companies that do a lot of prototyping and usability testing tend to evolve into an iterative process that uses small, frequent usability tests throughout the development cycle rather than one or two large efforts. Following are two examples.

From the Field: Iterative Usability Testing

"Although we do a lot of usability testing, we only bring in a few users each round. On the CP Select tool [described in Chapter 2], we tested five versions of the paper prototype, but we did our first two rounds of testing with two to three users each, and they were internal users. This works for us because our tools are highly specialized and we have ready access to domain experts in-house. For the next three rounds, we tested with both internal and external users, but only a couple of each. So for the whole project we worked with maybe six external users and a dozen internal. But that was enough to see radical improvements in our design—the team was surprised at how much easier it was for users to complete tasks in the final version.

"We also need to determine when we've done enough testing. What drives each round of testing is how clear we are on what the problems are. The data we collect also tells us what can stay the same. Over time, we can chart our progress by seeing how much of the design can stay the same—we call this 'gelling.' Once a design gels, we feel comfortable going off and implementing it, because the remaining issues tend to be minor."

By Jennifer Lymneos, The MathWorks

From the Field: Iterative Usability Testing—cont'd

"On one project I did usability tests every Friday for five weeks. I started with some rough prototypes our graphic designers had created in PhotoShop and printed them out. I sat down with four to six users (one at a time) on a Friday and walked them through a series of 'functional' questions (e.g., How would you use this page to do X?). I gave them markers so they could mark up the paper screen shots as they answered my questions. I spent the following Monday and Tuesday working my notes into design changes to apply to the PhotoShop files. In week 3 I began to show them the screen shots on a monitor—I'd gotten what I could from paper and was ready to switch over to doing it on a computer. By week 5 the users were looking at HTML files rather than flat graphic files (our designers were converting the flat files to HTML during this time). At the end of five weeks the design had stabilized to the point where I was only finding small issues."

By Brent Mundy, Usability Specialist

It Gets Easier

Your first paper prototype usability test is like busting sod—it's a lot of backbreaking work for a farmer to prepare a field for its first planting. Your first experience with paper prototyping may feel like a major production. You're starting from scratch with the user profile, recruitment, tasks, and prototype creation. Your co-workers have questions about the technique, and using a new skill takes a lot of mental energy. But for subsequent "crops," it gets much easier—you'll know where to find users, some of your materials can be reused, you won't have to explain the activities, you'll learn how to do the optimal amount of preparation, and you'll be more relaxed when conducting the usability tests.