# Chapter 22:
# PROTOTYPE TESTING

## Putting Your Ideas In Front Of Real Users

By this point you should know that I view the high-fidelity prototype as the primary means of describing the product to be built—a prototype is significantly more useful to the product team than the typical paper-based specification. However, that's really the secondary benefit. The primary reason to create a high-fidelity prototype is to help you gain a much deeper understanding of your product, and—ultimately—so that you can test your ideas with real users before you have your engineering teams take months to go build something that you have no real evidence will serve its purpose.

In this chapter I describe how to do this prototype testing. I'll warn you up front that this chapter is relatively long, but I will also say that *testing your ideas with real users is probably the single most important activity in your job as product manager.*

If your company is large enough to have its own usability testing team, by all means secure as much of their time for your project as you can. Even if you can't get much of their time, these people are terrific resources and, if you can make a friend in user research or usability testing, it'll be a huge help to you.

If your organization has funds earmarked for outside services, you may be able to use one of many excellent firms to conduct testing

for you. But at US$10,000-$20,000 per round of testing (typically around 10 users) that most of these firms charge, chances are that you won't be able to afford as much testing as your product will need.

If you're like most companies, you have few resources available, and even less money. But you can't let that stop you. It is absolutely essential that you test your ideas out with real users. As I said above, it is arguably the single most important part of your job.

So I'll show you how to do this testing yourself. Don't get me wrong, you won't be as proficient as a trained usability engineer, and it'll take you a few sessions to get the hang of it. But in most cases you'll find that you can still identify the serious issues with your product, which is what's important.

One thing to note is that, while usability testing (seeing if people can figure out how to actually use your product) is critical, you also need to test the value or usefulness of your product (do people actually want to use it?), and we'll discuss both forms of testing here.

## Finding Test Subjects

Before you do the prototype testing, you'll need to round up some test subjects. If you're using a lab they'll recruit and schedule the users for you, which is a big help, but if you're on your own, you've got several options:

- If you've established a charter customer program as I described in *Charter User Programs*, you should have quite a few users readily available. If you haven't yet established your program, then you should.
- If you're doing a product for business, then trade shows are a great source of target customers.
- It's increasingly common to advertise for test subjects on *Craigslist*. If you do this, try to keep your participant description a notch more general than specific, and then

when you call interested test subjects to explore their participation you can screen for the right match.

- For consumer products you can use your "friends and family" network, but try to avoid people too close to you, and those in the tech industry, unless that's specifically your target. Be sure to use subjects from outside this network, too.
- If you have a list of user e-mail addresses, you can do a selection from there. Often your marketing team can help you narrow down the list.
- You can solicit volunteers on your Web site—lots of major sites do this now. Remember: you'll still call and screen the people to make sure you don't get a sample skewed with early adopter types.
- One technique I especially like for medium to larger companies is to set up regular prototype test sessions—say every other Friday—where you arrange for 10-20 or so users to come into your offices for a couple hours each. Your product managers sign up for the time slots, so a given user might test a couple prototypes each. I like this because one person can do the logistics of invites and screening, and product teams can count on a ready set of test users on a steady basis.
- You can always take your show on the road and go to where your users congregate. If you're doing an e-commerce product, you may want to go to a shopping mall. If you're doing a sports product, go to a sports bar. If your product addresses a real need, you won't have trouble getting people to give you an hour of their time. Just bring some thank you gifts, and try not to look like you're trying to convert their religion.
- If you're asking users to come to your location—especially for business use—you will likely need to compensate the people for their time. If you're doing a consumer service product, a big sincere thank-you along with a hat with your

company logo on it will often suffice, as people genuinely want to help in the creation of products—especially for companies they like. However, if you do compensate test subjects, consider providing something along the lines of US$50 of credit on your site.

- Realize that there's a very high no-show rate when you schedule people to come in for testing—it's just a fact of life. While this number can rise to as high as 30%, you can drop it to the 5-10% range by giving your subjects a personal phone call the day before. Even leaving a voicemail message will help, but note that sending an email message does not work equally well.

## Preparing the Test

You'll need to define the usability tasks you'll want to test, and the interview questions concerning value:

- Define in advance the set of tasks you want to test. Usually these tasks are fairly obvious. If you're building an e-mail client, for example, your users will need to do things such as compose a message, read new mail, and file away messages. There will also be more obscure tasks, but concentrate on the primary tasks—the ones that users will do most of the time. If you have time, you can get to less common tasks but it's essential the key tasks are tested well.
- You have a one-time-only opportunity with each user you test—the opportunity to learn how they think about this problem today, without your product. If you're testing a new online restaurant rating service, rather than start them out at your prototype's home page, you might instead want to start them out with an empty browser and see what they do. What review sites do they use today? Do they use Google or Yahoo's search to find the specific restaurant, or do they go somewhere like OpenTable or Zagat? Do they search by neighborhood, by cuisine type, or price range? This type

of incredibly valuable information is missed if you jump right into your prototype, which will necessarily have quite a few assumptions built in. Once your test subjects have the opportunity to play with your prototype for a while, they can tell you what they like better, but they will no longer be thinking about the problem the way a first-time visitor would.

- You'll then want to get them to your prototype, but there's one more thing before you jump into your tasks. See if they can tell from the home page or landing page of your prototype what it is that you actually do, and especially what might be valuable or appealing to them. Again, once they jump into tasks, they'll lose that first-time visitor context, so don't waste the opportunity. You'll find that landing pages are incredibly important to bridging the gap between expectations and what the site actually does.
- After you've seen if your user can figure out how to do the tasks you're testing, now is the right time to have a conversation with him or her. Think of it as a one-person focus group. Does the person use a different product or site for the same purpose today, or is this something they do manually or offline? How much better is this than what they use today? And don't forget to ask my favorite question, Net Promoter Score (NPS): How likely would you be to recommend this product to your friends? Now that the user has interacted with your prototype, they understand the topic and you can have an extremely useful dialogue with them about this problem. Most importantly, you're trying to gauge how much this person values the product.
- It's useful if you structure your questions on a scale, such as 0-10, or with numeric answers in general. This is so that you can track the averages as they improve. One technique I like for gauging value is to ask how much the user would be willing to pay for it, even if you have no intention of actually charging for use this way. It's a way to assess value and—especially—to track how the average value goes up or

down over time as you change the prototype.

- Note that you don't have to wait until you have a complete prototype in order to begin testing. You can start with the main tasks, and it's okay if you have dead ends in the rest of the prototype. If the user wanders over to one of those dead ends, just ask "And what would you expect to happen if you did that?" This is a great question whether you have that path laid out or not. If you do have it laid out, you can see if they match. And if you don't, you'll get important info about what you'll need to do.

## The Test Environment

Here's how to prepare your test environment:

- Formal testing labs will typically have setups with two-way mirrors or closed-circuit video monitors, as well as cameras that capture both the screen and a frontal view of the user. Just know that while that's great if you have it, you do not need these things to have an extremely useful and valuable test. I can't count how many prototypes I've tested at a tiny table at Starbucks—just big enough for a laptop—with three chairs around the table. In fact, in some ways this is preferable to the testing lab because the user feels a lot less like a lab rat and may be more candid and open in his or her responses.
- The other environment that works quite well is your customer's office. It may be time consuming to go there and get set up, but even 30 minutes in their office will often tell you a lot. And because they are "master of their domain," they're frequently very open and talkative. Also, all the cues are there in the office to remind them of how they might actually use the product in their daily routine. You can also learn a lot from observing what the office looks like. How big is their monitor? How fast is their computer and network connectivity? How do they communicate with

their colleagues on their work tasks?

- There are tools for doing this type of testing remotely, but while you can see where their mouse is, and what the user is clicking on, it's not the same as looking at the person's eyes and body language. So, again, while more testing is generally better, this is not a substitute for face-to-face testing.
- As product manager, you need to make sure you are at every single test—do not delegate this task. Real value comes from experiencing as many users as possible—first hand—interacting with and responding to your ideas. Even if you use an outside firm to arrange and administer the tests, you need to be there with them during the testing. No one knows your product as well as you do, and you will have insights from watching the slightest hesitation or confused look, or the nuance of a question that reveals that your test subjects don't understand the conceptual model or particular feature. What gets summarized for you by a proctor will probably miss several key insights.
- Some people believe that the product manager (and the interaction designer) are too close to the product to do this type of testing objectively—that they may either get their feelings hurt or only hear what they want to hear. My view is that good product managers and interaction designers get past this very quickly. They know they will get the product wrong initially—that almost nobody gets it right the first time—and they know that learning from these tests is the fastest path to an inspiring product. So to me the benefits far outweigh the risks.
- Ideally, you should have one person administer the tests and another person taking notes. It's very useful to have someone to debrief with afterwards to make sure you both saw the same things and came to the same conclusions. That said, if it's just you and your laptop—and you've got a ready and willing target user in front of you—do it. It's all good.

- If you as product manager have a user researcher or usability engineer along with you, let him or her administer the test while you take notes. Otherwise, you'll probably be the one that administers. It's great to invite others on the product team to be your note taker. Most often this person will probably be the interaction designer, but the visual designer, managers, and especially engineers are all useful and they'll get a lot out of the experience.

## Testing Your Prototype

Now that you've got your prototype ready, you've lined up your test subjects, and you've prepared the tasks and questions, here are a set of tips and techniques for administering the actual testing:

- Greet the person warmly and offer a coffee or bottle of water, but the sooner you get to the prototype the better. Tell your subject you'll chat with them after they test the prototype, but you want to get their untainted impressions first. The more you chat beforehand, the more clues you are giving away and the less of a true first impression your test subject can provide. If more than five minutes goes by without the user starting in on the prototype you are talking too much.
- After your greeting, be sure to tell him or her that (1) this is just a prototype—it's a very early product idea—and it's not real, (2) they won't be hurting your feelings by giving you their honest opinion—good or bad, and (3) you're testing the prototype—you're not testing him or her. Your test subject can't pass or fail—only the prototype can pass or fail.
- When testing, you'll want to do everything you can to keep your users in "use mode" and out of "critique mode." What matters is whether users can easily do the tasks they need to do, and whether they value the product. It really doesn't matter if the user thinks something on the page is ugly or should be moved or changed. Sometimes misguided testers

will ask users questions like "What three things on the page would you change?" To me, unless that user happens to be an interaction designer, I'm not really interested in the answer to that question, or others like it. If users knew what they really wanted, software would be a lot easier to create. So watch what they do more than what they say.

- During the testing, the main skill you have to learn is to keep quiet. Normally, when we see someone struggle, most of us have an urge to help the person out. You need to suppress that urge. You have to turn into a horrible conversationalist and get comfortable with silence.
- There are three important cases you're looking for: (1) the user got through the task with no problem at all and no help, (2) the user struggled and moaned a bit but eventually got through it, or (3) the user got so frustrated he gave up. Sometimes people will give up pretty quick, so you may need to encourage them to keep trying a bit longer. But if the user gets to the point that you believe he would truly leave the site and go to a competitor, then that's when you note he truly gave up.
- In general, you'll want to avoid giving any help or "leading the witness" in any way. If you see the user scrolling the page up and down and clearly looking for something, it's okay to ask the user what specifically he's looking for, as that information is very valuable to you. Some people ask users to keep a running narration of what they're thinking, but I find this tends to put people in critique mode, as it's not a natural behavior.
- Act like a parrot. This helps in many ways. First, it helps avoid leading. If your test subject is quiet and you can't stand it any longer, tell them what they're doing: "I see that you're looking at the list on the right." This will prompt them to tell you what they're trying to do, looking for, whatever. If your subject asks a question, rather than giving a leading answer you can play back the question to them, "Will clicking on

this make a new entry?" The subject will usually take it from there because they'll want to answer your question: "Yeah, I think it will." Parroting also helps avoid leading value judgments. If you have the urge to say "Great!," instead say "You created a new entry." Finally, parroting key points also helps your note taker because he or she will have more time to write down important points.

- Fundamentally, you're trying to get an understanding of how your target users think about this problem, and to identify places in your prototype where the model the software presents is inconsistent or incompatible with how the user is thinking about the problem. That's what it means to be counterintuitive. Fortunately, when you spot this it is usually not hard to fix, and can be a big win for your product.
- You will find that you can tell a great deal from body language and tone. It is painfully obvious when test subjects don't like your ideas, and it is also clear when they genuinely do. If they like what they see, they'll almost always ask for an email telling them when the product is out. And if they really like it, they'll try to get access from you before it's released to the general public. When I attend the prototype testing with my clients in Germany, even though I don't speak German, it is obvious what the issues are—which ideas work well and which ones don't.

## Updating the Prototype

The point of this prototype testing is to identify what you need to fix in the prototype to make it more valuable and usable. So, as quickly as possible, you'll want to correct the problems.

- Some people believe you have to freeze the prototype, the tasks, and the questions for a complete round of testing (generally 6-8 users) before drawing any conclusions. I don't think that's true. I have found that you can significantly

accelerate the process of getting to a good product by responding more quickly to feedback.You don't have to be hit on the head by eight users in a row to know you need to fix something big. So go ahead and fix it when you feel you've identified a problem, even if it's after only two or three users. The harder question is deciding when you're done. Generally, if you can get through six consecutive users who understand and appreciate the value of the product—and can get through the key tasks—you're in good shape and you've done your job.

- You might determine that you just aren't able to get people interested in this problem, or figure out a way to make the product clear or usable enough that your target users realize its value. In this case, you may decide to stop right there and put the idea on the shelf. Some product managers consider this a big failure. I view it as saving the company the wasted cost of building and shipping a losing product, not to mention the opportunity cost of what your engineering team could be building instead.

This whole process might sound complicated or difficult, but the remarkable thing is just how easy and effective it actually is. The best way to prove this to yourself is to take your laptop with your product or prototype on it to someone that hasn't seen it yet and just give it a try.

I want to give credit here to two important sources (and great resources for you).

The first is the book, *Don't Make Me Think* by Steve Krug. It's mostly a book on interaction design, but in the back third he makes a compelling case for this sort of informal testing, and he provides many useful tips. I have long recommended this book to both product managers and designers, and I hope you'll buy a copy and read it carefully.

Second, my favorite product testing firm is Creative Good (www.creativegood.com). These guys specialize in a form of testing they call *Listening Labs*, which is a powerful form of undirected testing that can find huge issues with your product—functionality and design—resulting in dramatic improvements to the business results. While most testing firms test the products on a task basis, these guys are good at looking at the big picture, which is where many of the biggest improvements are to be found. Several of the techniques described above are adapted from their testing methodology.