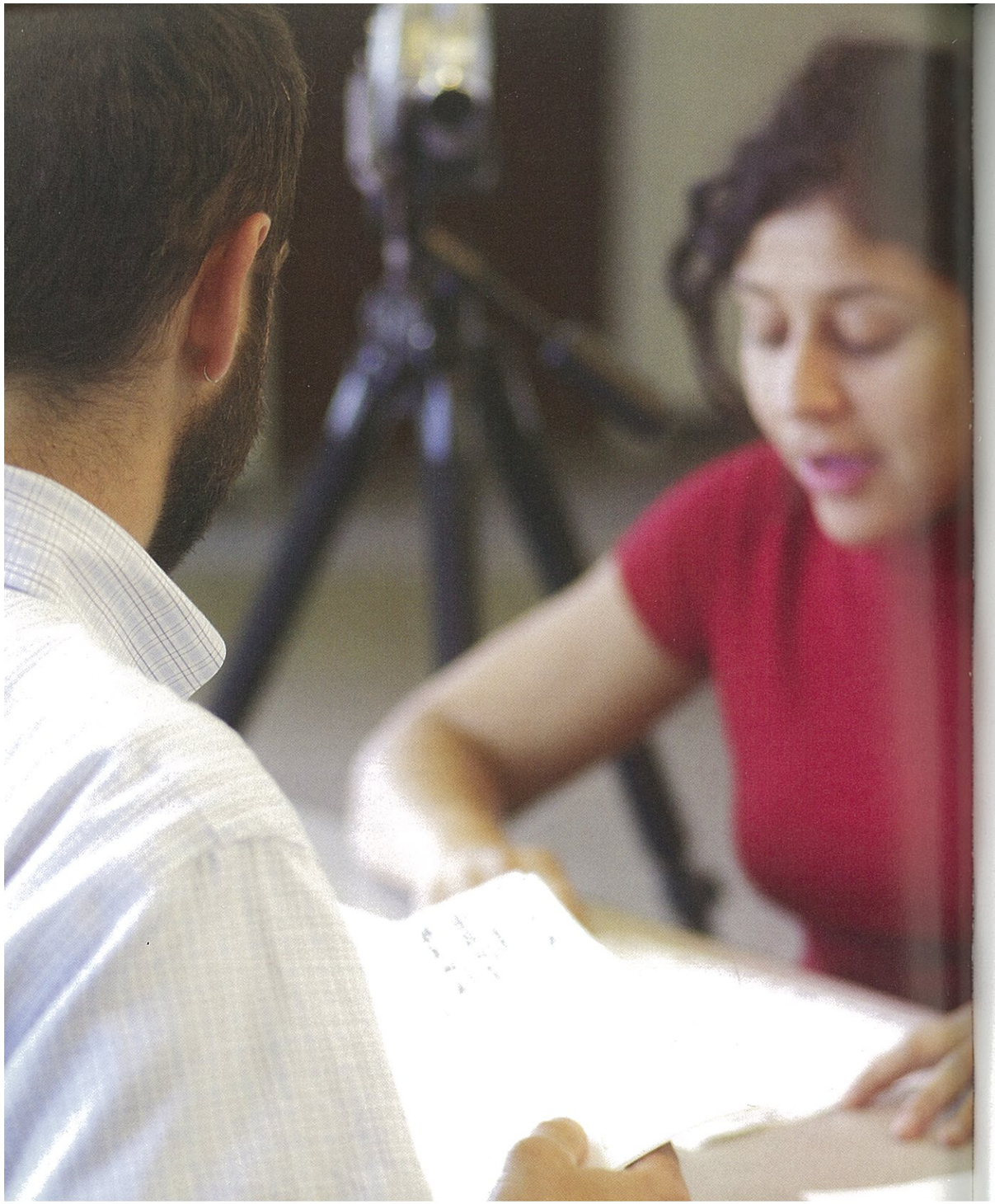

CHAPTER 23

Evaluating Your Design

In my experience, the methods described in this book yield consistently good results when applied by skilled practitioners. However, it's unlikely for even the best designer to be exactly right about everything. Collaboration with other designers, design reviews with a team lead, and reviews with engineers, subject matter experts, and stakeholders provide a form of ongoing design evaluation throughout any project. However, it's a good idea to do a more formal type of evaluation at least once before you send your product out into the world. Think of this as the designer's equivalent of QA.

One chapter isn't enough to do any evaluation method justice; my intentions here are merely to introduce the topic, to demonstrate how design evaluation and common evaluation methods fit within the context of Goal-Directed Design, and to help you avoid a few of the most common pitfalls.

Why, When, and What to Evaluate

When and how you evaluate your design depends on why you're evaluating it. Design evaluation can serve several purposes:

- **Persuading people there's a problem.** If you're the only one who's convinced the existing product or design direction needs work, an evaluation can be an effective tool for changing minds.
- **Improving design.** Some types of design evaluation can help you see if you've designed the right product, while others can help you see if you've designed the product right.
- **Helping designers choose between two approaches.** Personas, scenarios, and a collaborative approach usually point clearly in one direction. If that's not working, you can use an evaluation to help you decide which approach is better.
- **Demonstrating design's effectiveness.** If you're convinced your design is right, an evaluation can help show stakeholders what a great job you're doing—and if the evaluation isn't so good after all, you'll learn something useful.
- **Gathering kudos for marketing.** Taste tests aren't just for soft drinks. This one isn't usually a designer's concern, however.

he one thing design evaluation of any kind oesn't do is *generate* good design. Some peole expect usability testing, in particular, to be complete solution: Prototype your best guess, est it, then keep tweaking and testing until you et it right. This is a bit like living on ice cream nd potato chips and expecting your doctor to fix ou with a pill; if I can repurpose the cliché, an unce of good design is worth a pound of evaluaon. This is why I always advise clients with tight udgets to forgo testing in favor of more up-front esearch and design time. However, I would ever advise skipping evaluation for any product r service where a usability problem could be isastrous, such as for a medical device or vecle interface.

When to evaluate your design depends on what ou want to accomplish. A **formative** evaluation elps you know whether you're on the right path. may only focus on a single interaction, so you an do this kind of evaluation anywhere along he way. Although many designers find formative valuations helpful, I've found that researchased personas, scenarios, and experience ttributes, along with good collaboration, have ever left me in doubt about how to proceed. A **ummative** evaluation is meant to help you polsh odds and ends or persuade people about the esign (one way or the other). This is most efective when you have a complete or nearly comlete design, usually once you've finished docuenting the first draft of your interaction design r, for hardware, when you have an appearance odel. A **comparative** evaluation that pits two or ore products or concepts against one another ould be either formative or summative.

What to evaluate depends on the product or serice as well as your own confidence in the deign. If you're uncertain as a team about a parcular design decision, you'll want to evaluate at specific decision. For hardware, it's worth ssessing whether the product and its controls work ergonomically for the target audience. For most products, you'll want to evaluate whether you made any mistakes in general, especially in the most important areas of the product; tiny flaws in obscure corners aren't always worth looking for.

Types of Evaluation

There are several useful ways to evaluate design. Which approach works best depends on your timeline, budget, and—most important—your objective. For identifying usability issues and evaluating functional design directions, usability testing, expert reviews, and discussions with individual users are the methods of choice. Focus groups and individual user discussions are common approaches to assessing aesthetic impact.

Focus groups

I've worked with clients who believe strongly in focus groups (or individual interviews along the same lines) for predicting how well a design concept or design language will be accepted. Let me be blunt: Focus groups are pretty much useless for assessing interaction design because until someone interacts with the product instead of just looking at it, their opinions are uninformed at best. Even for assessing design language, I must say I'm not a fan of the approach.

Why not? Focus groups are easy to do badly (see Chapter 9). Far too many people take focus group results as gospel rather than as one of many data points to consider; I'm always concerned by the idea of letting users decide how a brand should be represented. Also, what people are drawn to in a focus group can't really predict what they'll be drawn to in a store jammed with products; self-reported behavior often has little to do with reality.

If you are working with people who insist on doing a focus group, you should at least argue for not asking people what they *like*. Instead, consider using the experience attributes to drive the discussion. Which of these words best describes the design: powerful, simple, smart, or friendly? Which design is the most elegant? Take these as input about whether you've accomplished your design intent, not as guidance on what decision you should make.

What people are drawn to in a focus group can't really predict what they'll be drawn to in a store jammed with products.

Expert reviews

In an expert review, an experienced design or usability professional steps through the product or design looking for likely problems and evaluating their severity. Expert reviews are generally quick and inexpensive. They have fallen somewhat out of fashion because they're based on the opinions of an individual, and are therefore seen as less scientific than other approaches. Although it's certainly true that expert reviews only work when done by someone who actually is an expert, there's nothing wrong with relying on expert opinion: People do it in law, engineering, and many other fields.

However, unless the reviewer has considerable experience in your product's particular domain, an expert reviewer can usually identify only the issues that violate broadly applicable design principles, not those based on flaws in domain-specific workflow. For this reason, an expert review is most effective when combined with a day or so of field research, or at least a couple of hours discussing users and typical scenarios with subject matter experts. In my experience, an expert review does a better job of identifying and prioritizing issues when it involves some consideration of user goals and workflow in addition to design principles.

If you're hiring an expert to review your work, she will typically offer anything from a day of live discussion to a few days with a written report and possibly recommendations for adjustments. Take a look at an example report and see if the expert describes the basis for his assessment (i.e., ties each critique to design principles or experience with relevant users) and distinguishes disasters from nits. A helpful expert review may also point out things that *might* be issues, even if the reviewer is uncertain—for example, "The sequence of fields here seems odd because it differs from the mental model I would expect these users to have. However, since I haven't interviewed any users, this assessment may be incorrect. Consider having some users look at it."

gorous though
od usability
sting methods
e, various
udies show
ey're not the
olproof science
any people
elieve them
be.

Usability testing

In a usability test, individual users work through a series of fairly realistic tasks. Some tasks may be timed, but most often, each participant talks out loud to describe his thought process as he uses an actual product or prototype. Many people believe this approach is both more objective and more effective at identifying issues than an expert review, so testing has become a sort of gold standard in design evaluation. It's also tremendously persuasive; if five out of ten users couldn't accomplish a task, few people would doubt that there's a problem.

Rigorous though good testing methods are, however, various studies show they're not the foolproof science many people believe them to be. For example, since 1998, Rolf Molich and a number of his colleagues have conducted a series of seven studies called CUE: comparative usability evaluations. In each, they've asked a set of experienced teams to evaluate the same product either employing testing or expert review techniques. While each study had a slightly different focus, the results have consistently shown that a usability test doesn't find every problem, and that tests conducted by different people find different results. In the CUE-1, CUE-2, and CUE-4[1] studies, for example, anywhere from 60 percent to 91 percent of the usability issues were reported by only one team, and many of these were severe issues that resulted in failure to complete tasks. Of the 340 usability issues reported in the CUE-4 study, only nine issues were common to more than half the teams. Jacobsen, Hertzum, and John[2] found a similar phenomenon in a 1998 study in which four HCI experts reviewed a video of exactly the same test. The experts again offered divergent analyses, with only one evaluator reporting 46 percent of the issues. In other words, the effect of the evaluator is substantial in anything but a purely quantitative study.

High tech methods, such as eye tracking and usability labs full of equipment, promise a more objective approach, but the results are questionable. Objective comparisons of task time with one design versus another are hardly the only measure of effective design. In the as-yet-unpublished CUE 7 study[3], Molich found that eye tracking did not identify any issues that weren't already identified using less expensive

Molich, R., and Dumas, J.S. "Comparative usability evaluation (CUE-4)." *Behaviour & information technology*, Vol. 27, issue 3, 2008.

Jacobsen, N.E., Hertzum, M., and John, B.E. "The evaluator effect in usability studies: Problem detection and severity judgments." Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting, October 1998.

methods. While his small sample makes the results inconclusive, they make sense—eye tracking can tell you, for example, whether a participant is reading or scanning, but it can't tell you whether she's actually absorbing what she sees. Expensive labs and tools are useful for fundamental HCI research, but for evaluating products, you're probably better off using less expensive techniques like those shown in Figure 23.1.

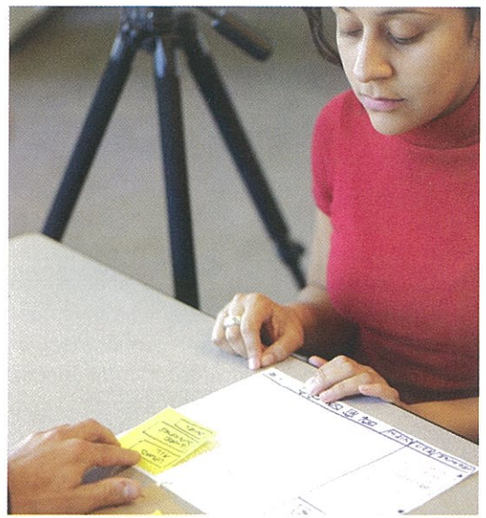

Figure 23.1. An inexpensive usability test using a paper prototype and basic camcorder.

So, what should you take from all this? There's no foolproof way to evaluate a design's effectiveness; the people evaluating designs are no more perfect than the people creating them, and even the world's best design could fail in the marketplace if it's priced incorrectly or marketed poorly. That doesn't mean you shouldn't evaluate your design, but it does mean that the more important it is to get the design right, the more evaluation techniques and evaluators you should use.

PLANNING A TEST

Planning a test consists of identifying what you want to learn and from whom, coming up with some tasks that will help you learn it, deciding how to describe those tasks to users, and determining what level of prototype fidelity you'll need.

Deciding what you need to learn

Do you need a quick answer to a self-contained problem or two, or do you need a comprehensive hunt for problems in your design? Your answer (and, yes, your budget) will determine how thoroughly you'll need to test. Quick answers are easy enough to find in a couple of days. A thorough test of multiple tasks can require a month or more to plan, execute, and evaluate. The first sort is usually easy enough for a design team to squeeze in unless there's no flex in the schedule; the second probably calls for a dedicated usability tester to handle the bulk of the work.

Identifying participants

Determining the number and type of test participants you need is a lot like planning your user interviews (see Chapter 6). Unless you have questions involving a specific user population—such as whether your design is as accessible as you think it is—it's usually fine to use the same recruiting criteria. As in field research, small samples of four or five people are fairly effective for narrowly defined roles, whereas you'll want a larger group of 15 to 20 for applications and Web sites with diverse audiences.

Of course, there's also the truly low-budget, informal version of recruiting participants: Waylay your colleagues in the hall or tell them to stop by for a snack and walk through a task. This only works well if your colleagues are reasonably similar to end users, however.

systems
d by experts,
not particularly
ortant—and
be counter-
ductive—
st ease
arning;
iency over
is more
ortant.

Determining your focus

One of the most important decisions you need to make is whether you're interested in ease of learning or efficiency of use over time. It's reasonable to assess whether a passenger can walk up to an airport kiosk and use it right away, but it's not particularly important—and may even be counterproductive—to test whether an air traffic controller can immediately understand how to use his complex system without training.

When efficiency of use for intermediate or expert users is the more important metric, you need to include some form of pre-test training before you jump into test tasks. At minimum, this can consist of a quick walkthrough of the prototype and perhaps a handout that identifies controls by function. For greater realism, you would need to assess how users did after working with a functional system for a while, such as in a limited beta release.

Designing tasks

As Carolyn Snyder says in her very useful book, *Paper Prototyping*,[4] an effective task:

- Is based on user goals
- Is related to issues of importance for product success
- Has an appropriate scope (not tiny, but finite and manageable)
- Has a limited and predictable set of possible solutions

As it turns out, the tasks that form the basis of your key path scenarios (see Chapter 16) generally meet all of these criteria and are a perfect starting point for most tests. Better yet, if you're at the point where you've got a first draft document, you already have most of what you need for a simple paper prototype study.

Deciding what kind of prototype to use

A prototype can be either high or low **fidelity**: either faithfully rendering the eventual user experience or only approximating it. A high-fidelity appearance model is very similar to the mass, finish, and (rarely) some of the mechanical functionality of the final product. A high-fidelity software prototype is most likely clickable, with realistic

yder, C. *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann

data entry and renderings of animated behavior, such as button clicks. Good fidelity is important for testing complex or subtle interactions, such as a dense information display with a lot of direct manipulation. High-fidelity prototypes can frustrate users if their fidelity isn't quite high enough, however; if it looks like a real system, users tend to expect that every button works and is as responsive as production code.

Low-fidelity prototypes, such as the sketches on paper shown in Figure 23.2, are quick and inexpensive to produce, and they don't set unrealistic user expectations. Unfortunately, they can introduce a certain type of error simply by being so unrefined. For example, a prototype without thoughtful visual design is probably missing essential clues about hierarchy, which would make it more difficult for a participant to pick out what's important on the page. Typical wireframe conventions, such as using a box with an X to represent an image, often don't make sense to users, either. This may not be critical with a simple, form-based screen but is essential for many rich data displays.

The happy medium for most software is a paper prototype based on your detailed screen drawings, like the touch screen prototype shown in Figure 21.28. This allows fast production, especially since you probably have most of the screen states you need drawn for scenarios, but still gives users the benefit of clear hierarchy, clean layout, and any rich visual feedback.

Although you can get feedback on future product concepts even with sketches, many systems involving hardware are best tested with higher-fidelity prototypes. For assessing interaction, these don't need to have realistic surfaces. However, they should use the correct controls in a realistic relationship to a display of appropriate size and resolution, and should be positioned in the way they'll eventually be used. For example,

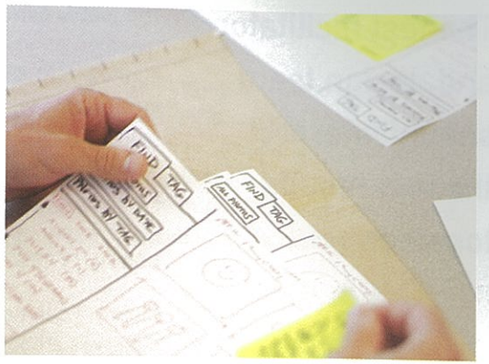

Figure 23.2. A low-fidelity paper prototype.

you could mount a touch screen and number pad for an automatic teller on a piece of acrylic, then mount the acrylic to a wooden frame at the correct height and angle.

Whatever you do, don't assume the low fidelity of a prototype means the design can be half-baked; even if you're using the lowest degree of fidelity, the text and widgets on the screen or the function of hardware controls should still be thoroughly considered and expressed.

WHO SHOULD FACILITATE A TEST AND INTERPRET DATA?

While the design team should always be closely involved in planning a test, it's not ideal to facilitate your own tests. It's difficult to maintain objectivity; you may find yourself leading the witness or providing excessive coaching. However, it's better for the design team to conduct the test than for someone like a product manager—who is probably less knowledgeable about testing and equally likely to have biases—to do it. Bring in an outside tester if you can, even if it's another member of your design group who hasn't worked on this particular project.

ıe test facilitator ıd design team ıould interpret e test results gether.

The test facilitator and design team should interpret the test results together. An expert facilitator knows how to read user responses during a test, but the design team's field-research experience may provide explanations for some responses. For this reason, I don't recommend having a third party test your design and hand the report to management without your involvement; some unfortunate misunderstandings might happen.

USABILITY TESTING RESOURCES

Clearly, I can't do usability testing justice as part of a brief chapter; it deserves a book all its own. Fortunately, there are some good ones out there. While you'll need to do some interpretation to see how the methods fit within the design process described in these pages, you'll find a wealth of information on planning, moderating, and interpreting tests in Carolyn Snyder's book, as well as in *The Handbook of Usability Testing*,[5] Jeff Rubin's popular book, which he and Dana Chisnell recently updated.

Comparative evaluations

For a comparative evaluation of any sort—whether usability test, focus group, or expert review—the biggest potential pitfall lies in the fidelity of the things being compared. Comparing an incomplete design to a finished product introduces bias in both directions. A finished product may fare better simply because it is more polished. On the other hand, a low-fidelity prototype might not get as much critique simply because there's not that much to comment on. Expert reviews are less prone to this sort of problem than direct user feedback, but even experts may succumb.

The best approach is usually to dumb down the real product to the same level of fidelity as your design. If you're using a paper software prototype, create similar sketches of the real product as your basis for comparison. If you're comparing a foam or appearance model to an existing physical product, you might have to disable controls, weight the foam model so one doesn't feel more "real," or paint the real product a flat gray to match your model.

Rubin, J., and Chisnell, D. *The handbook of usability testing: How to plan, design, and conduct effective tests.*

Summary

Just as every author needs an editor, every designer can benefit from having someone evaluate the effectiveness of her design. Usability testing and expert review are no more a science than design is, but both offer a rigorous approach and some tried-and-true techniques. Focus groups, on the other hand, yield inconsistently reliable data at best. If it's at all possible, you should build a test or expert review into your project plan. That said, if you would have to shortchange design or initial research to do it, focus on the activities that prevent problems, rather than trying to make up for a rushed effort with some last-minute QA.