

# Chapter 1

## Introduction

Paper prototyping is a widely used method for designing, testing, and refining user interfaces. In the early 1990s it was a fringe technique, used by a few pockets of usability pioneers but unknown to the vast majority of product development teams (and often considered pretty darn weird by the rest). But by the mid-1990s, paper prototyping was catching on. People at well-known companies (IBM, Digital, Honeywell, and Microsoft, just to name a few) experimented with the technique, found it useful, and started using it as an integral part of their product development process. As of 2002 paper prototyping is not considered nearly so weird, and the technique is mainstream practice at many companies, both large and small. There are, however, still many people who've only heard enough about paper prototyping to be intrigued—this book is for you.

For much of its history, paper prototyping has been a tool clenched firmly in the hand of the academic researcher or usability specialist. Like any useful tool, though, its greatest potential can be realized by placing it in the hands of nonspecialists along with instructions for its proper use. I believe that anyone who is involved in the design, implementation, or support of user interfaces can benefit from paper prototyping because it fosters development of products that are more useful, intuitive, efficient, and pleasing. Although you can't learn everything about a topic from one book, this one gives you enough knowledge about paper prototyping to start using it.

### What Is Paper Prototyping Anyway?

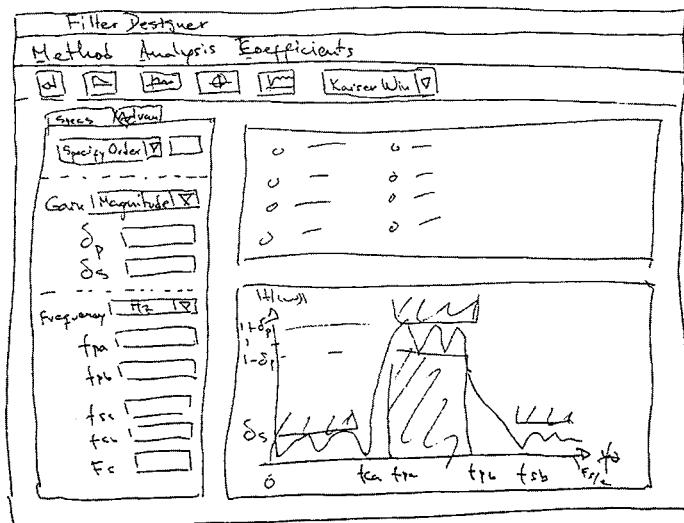
In its broadest sense, paper prototyping can be considered a method of brainstorming, designing, creating, testing, and communicating user interfaces. This

book emphasizes the creating and testing aspects of paper prototyping, although I touch on the others as well. The technique is platform independent and can be used for Web sites, Web applications, software, handheld devices, and even hardware—anything that has a human-computer interface is a potential candidate for paper prototyping.

I'm not aware of any official definition of paper prototyping, and I've heard people use the term in reference to several different methods. Here's the definition of paper prototyping I use in this book:

*Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person “playing computer,” who doesn’t explain how the interface is intended to work.*

Here's how it works: You meet with other members of your product team to choose the type of user who represents the most important audience for the interface. You determine some typical tasks that you expect this user to do. Next, you make screen shots and/or hand-sketched versions of all the windows, menus, dialog boxes, pages, data, pop-up messages, and so on that are needed to perform those tasks. It is not necessary to have a working version of the interface. If you can sketch it on a whiteboard, you can make a paper prototype of it. Figure 1.1 shows an example of a hand-drawn paper prototype screen.



**Figure 1.1** A hand-drawn paper prototype of a screen from an application used to design filters for scientific data.

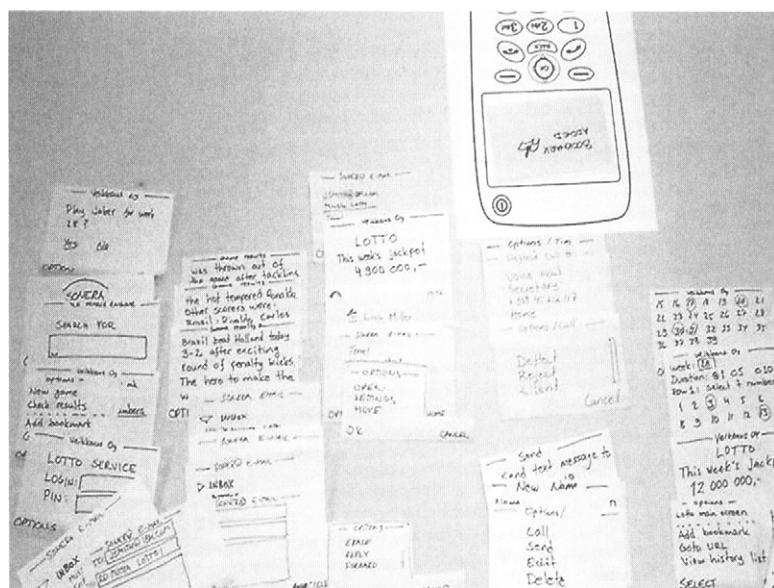
After you create the prototype, you then conduct a usability test. You bring in a person who is representative of the audience you and your team members agreed on. You ask this **user** to attempt the tasks by interacting directly with the prototype—“click” by touching the prototype buttons or links and “type” by writing data right on the prototype. One or two of you play the role of “**Computer**,” manipulating the pieces of paper to simulate how the interface behaves but without explaining how it is supposed to work. A **facilitator** (usually someone trained in usability) conducts the session while other members of the product team act as note-taking **observers**.

You will quickly discover which parts of the interface work well and which are the trouble spots. Because the prototype is all on paper, you can easily modify it right after—or sometimes even during—each usability test. You can conduct several usability tests in just a day or two, and it doesn’t take long to see the patterns in the feedback you’re getting. Thus, paper prototypes allow you to iterate and improve a design quite rapidly based on input from real users, and this can all happen before the first line of interface code is written.

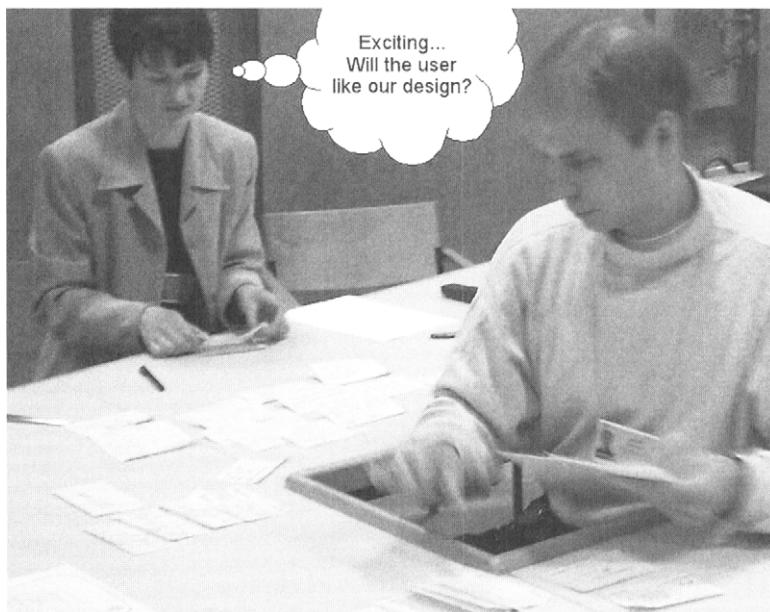
The previous discussion makes reference to four roles: user, facilitator, Computer, and observer. Figures 1.2 to 1.8 show these four people in action. (With the exception of the facilitator, there can be multiple people in each role, especially observers. So this is a minimalist example, but still a realistic one.)



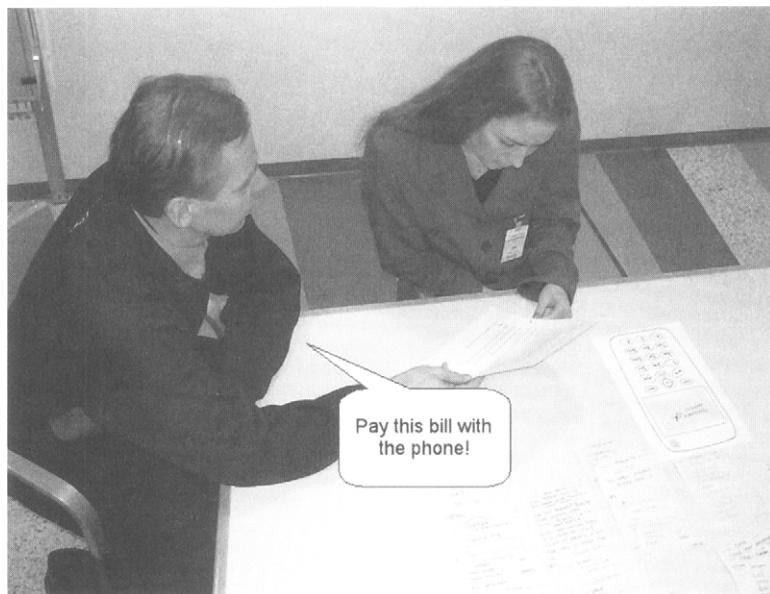
**Figure 1.2** Paper prototyping is a team effort. After creating the usability tasks, the product team works together to generate a mock-up of the interface.



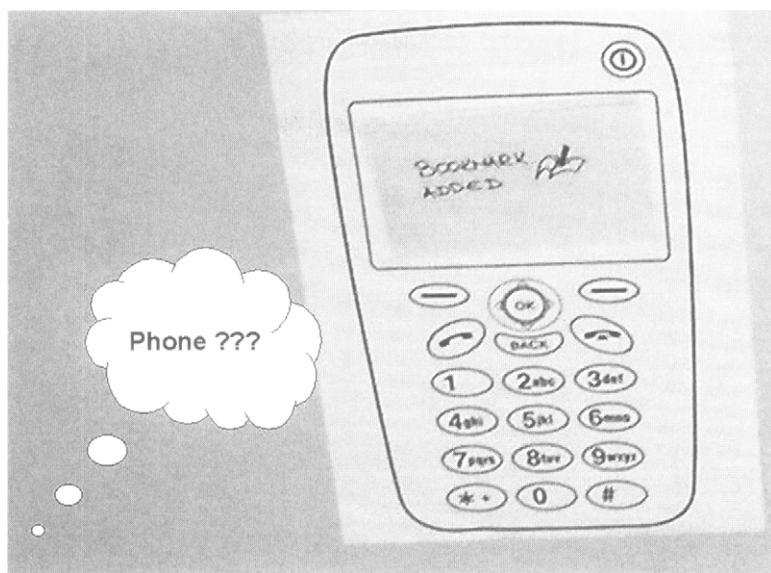
**Figure 1.3** Individual pieces contain content that is relevant to the usability tasks—in this case, using a wireless phone to check lottery results and sports scores.



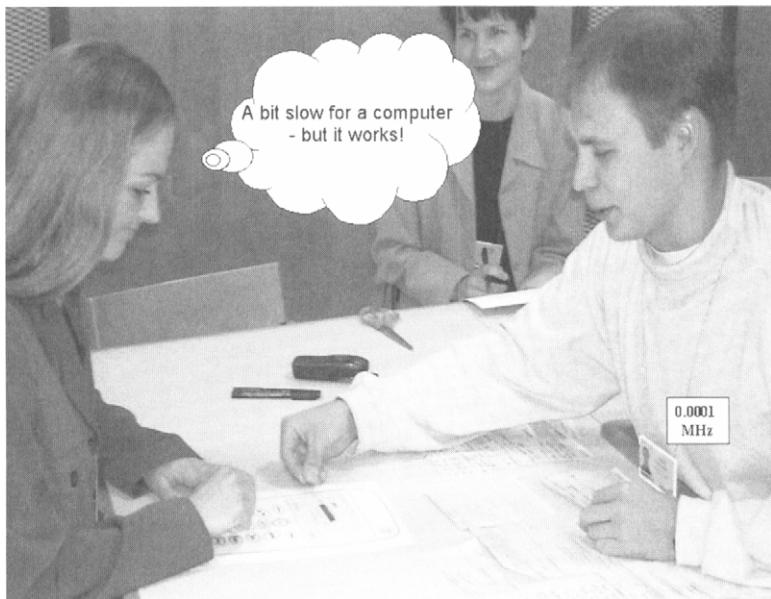
**Figure 1.4** The Computer practices the tasks before the first user arrives. Another team member reviews her list of issues that she hopes the usability tests will cover.



**Figure 1.5** The facilitator explains the purpose of the session and how to interact with the prototype. The facilitator sits next to the user, giving each task and interacting with her as needed.



**Figure 1.6** The user might find paper prototyping odd at first but quickly gets into the spirit.



**Figure 1.7** The Computer highlights the item the user has just “clicked” on. Other team members observe quietly and take notes. The facilitator (not visible) is still sitting next to the user.



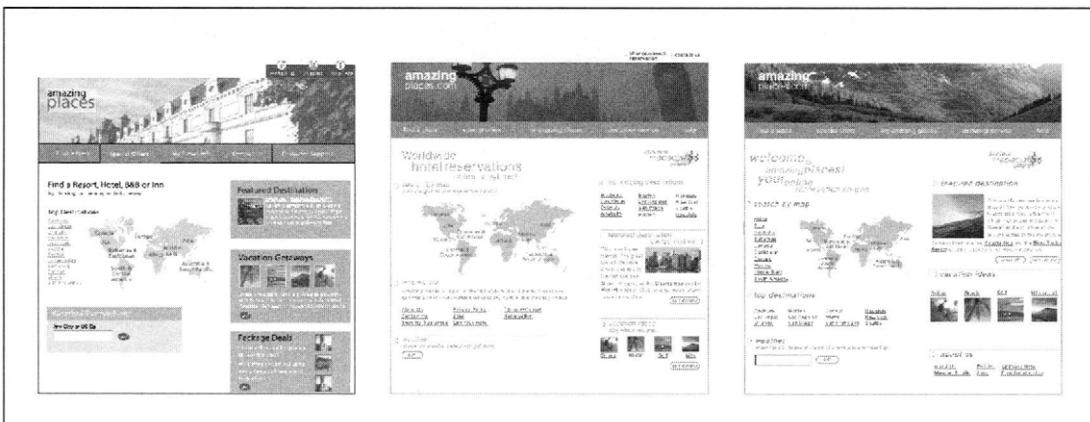
**Figure 1.8** Paper prototyping is a creative activity and is often quite fun.

## What Paper Prototyping Isn't

There are three techniques—comps, wireframes, and storyboards—that people commonly confuse with paper prototypes. These techniques are useful, but they usually don't fit my definition of a paper prototype, although all of them can be turned *into* paper prototypes. Here's a bit more explanation.

### Comps

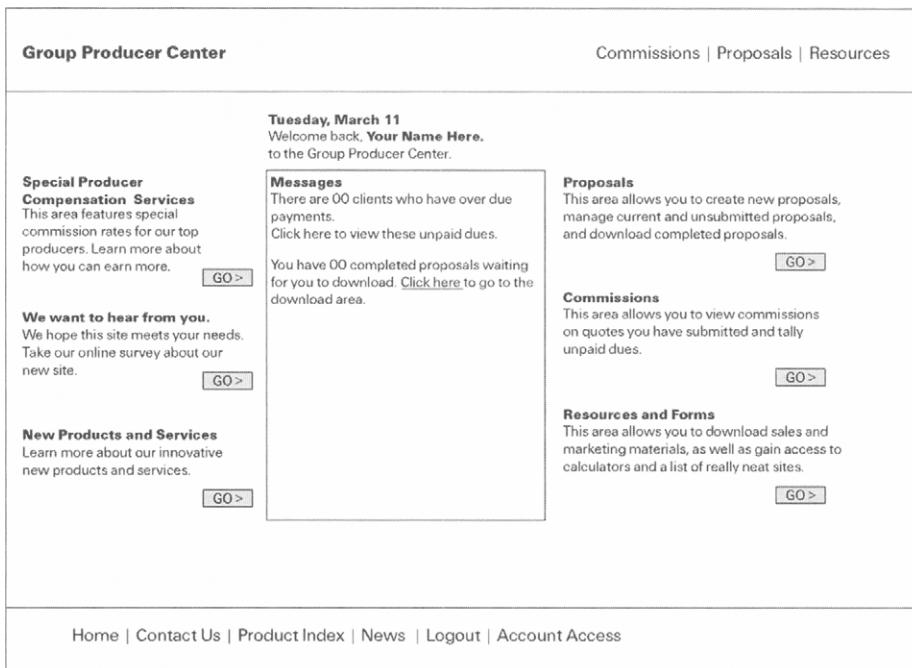
Comps (which is short for *compositions*) are visual representations—usually of a Web site—that show the look of the interface, including colors, fonts, layout, logos, artwork, and so on. (Figure 1.9 shows a sample of some comps.) The graphic designer or agency responsible for the visual aspects of the design might make several variations of the interface, allowing the decision makers to pick the one that best supports the current business initiatives, conveys the brand, and so forth. Some comps use nonsense words to represent the text and links. Comps are primarily used in internal discussions of a site's visual design; they usually are not intended (or suitable) for usability testing because users can't interact with them. However, if comps contained realistic content and were printed out, they might then fit my definition of a paper prototype.



**Figure 1.9** A set of comps for the home page of PlacesToStay.com. Comps are used to explore different layouts, graphics, and visual emphasis. Unlike this example, some comps use nonsense words for text and links.

## Wireframes

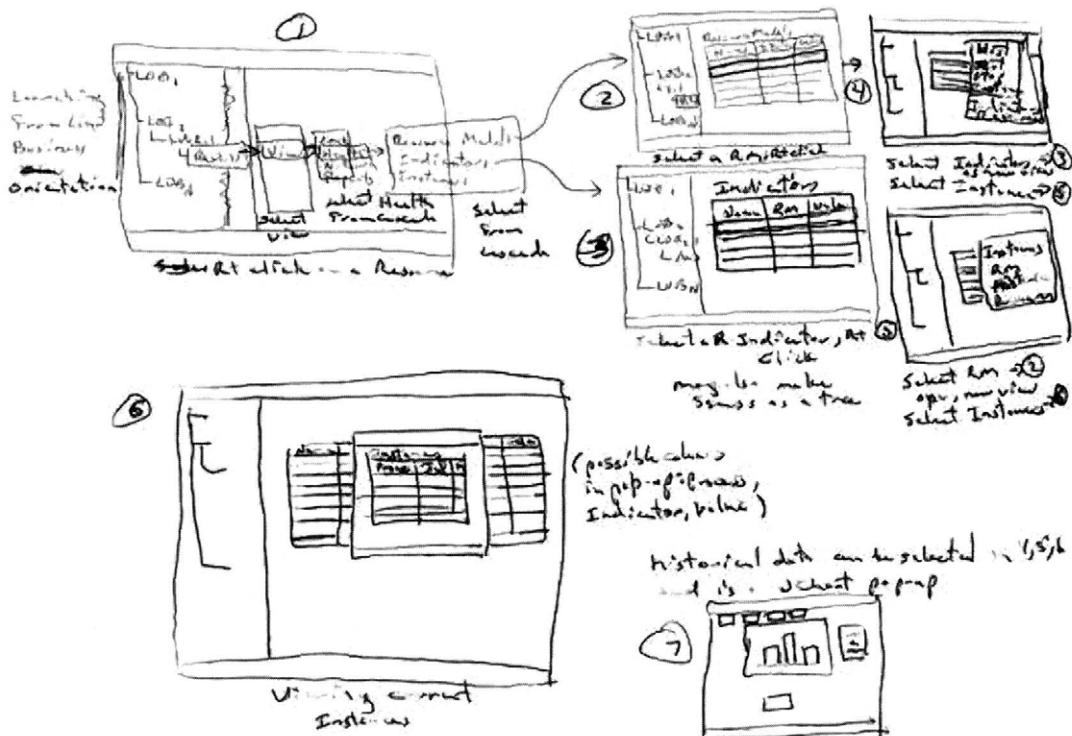
Like paper prototype, *wireframe* can be a confusing term because people use it to mean different things. A wireframe defines the page layout for a Web site, showing what content goes where. (Figure 1.10 shows an example of a wireframe.) In the early stages of designing a Web site, wireframes are used in determining the page layout and navigation. But is a wireframe a paper prototype? It depends. Some wireframes designate the major areas on the page with labels (for example, “product information”) but don’t contain any content. This type of wireframe is sometimes used to get feedback from users, but this approach is of limited benefit because it’s hard to tell whether the user’s understanding of “product information” is the same as the designer’s. Thus, a wireframe without content doesn’t quite fit my definition of a paper prototype. On the other hand, with the addition of realistic content a wireframe could be printed out and tested as a paper prototype. In that case I would classify the wireframe as a paper prototype.



**Figure 1.10** A wireframe shows the layout of a Web page, but often the content is represented by nonsense words.

## Storyboards

A storyboard is a series of drawings or images that represents how an interface would be used to accomplish a particular task. It's basically a flowchart. Some storyboards, like the one in Figure 1.11, include representations of the user interface, but other storyboards are more conceptual and high-level. As the name implies, storyboards are often spread across a wall. They are typically used to understand the flow of the user's work and how the interface will support each step. Storyboards are most often used within the development organization, although sometimes users review them. Because users can't interact with storyboards (they can only look at them), I wouldn't classify them as paper prototypes. However, you could readily turn a storyboard into a paper prototype by taking it down from the wall and adding whatever data is needed to support a task scenario.



**Figure 1.11** A hand-sketched storyboard used by developers to work out a sequence of screens. The annotations explain choices the user makes or processing done by the system.

## Benefits of Paper Prototyping

Here's a preview of paper prototyping's advantages:

- ◇ Provides substantive user feedback early in the development process—before you've invested effort in implementation.
- ◇ Promotes rapid iterative development. You can experiment with many ideas rather than betting the farm on just one.
- ◇ Facilitates communication within the development team and between the development team and customers.
- ◇ Does not require any technical skills, so a multidisciplinary team can work together.
- ◇ Encourages creativity in the product development process.

Someone once asked me what the paper prototyping “bumper sticker” would say, and my answer was, “Maximum Feedback for Minimum Effort.” That’s really what it boils down to—an efficient means of getting make-it-or-break-it information about your interface. Using only a few office supplies and a dash of ingenuity, you can get all sorts of useful feedback in time to do something about it before the next release.

Of course, no technique is perfect, and this includes paper prototyping. One important drawback is paper prototyping’s difficulty in detecting some classes of problems. In addition, depending on the circumstances of the project, there are cases in which the benefits of paper simply aren’t very compelling.

## Paper Prototyping and Usability

Entire books are devoted to usability (a.k.a. user-centered design). Although I can't summarize the entire discipline in a few paragraphs, I've found it helpful to think about usability in the following ways:

- ◇ The goal of any user-centered activity is to make the interface better for its intended audience and purpose, in a way that is consistent with the business goals of the organization producing that interface.

- ◇ Usability is like love: The more you give away, the more you have. You can help spread the love by passing along the concepts in this book to others. It'll come back to you in the form of more successful products.
- ◇ Usability is also similar to pornography in its ability to elude precise definition. To paraphrase the famous line by U.S. Supreme Court Justice Potter Stewart, "I know usability when I see it." If you do much reading about usability, you'll come across many definitions of it (some more usable than others). Don't get bogged down in the semantics; you'll be able to recognize good usability before you can define it.
- ◇ Like Don Norman says, a usable interface becomes invisible; sometimes you know you've gotten it right when your customers/users *don't* talk about how usable the product is . . . they're too busy raving about how you've made their life better.

Don Norman is a pillar of the usability profession. His first book, *The Design of Everyday Things*, is 15 years old but still a classic. It's a deceptively entertaining read that will have a profound effect on how you look at objects and interfaces of every kind. Don first described the concept of an invisible computer in DOET (as it's affectionately abbreviated) and later went on to write a whole book called *The Invisible Computer*.

Readers familiar with usability may notice that I tend to view paper prototyping and usability testing as overlapping and often synonymous concepts. But that's a generalization, and like any generalization it isn't entirely accurate. Many companies create prototypes in software and conduct usability tests with them. Paper prototypes also have uses beyond usability testing, such as internal reviews. But most of the time when I use the term *paper prototyping*, I assume that the prototype is being created with the intent of usability testing it.

## The History of Paper Prototyping

I've been using paper prototypes since 1993, but I didn't invent the technique. Neither did Jared Spool, whom I learned it from during my consulting days at User Interface Engineering. But determining where paper prototyping originated is like trying to track a river upstream to a single source. The best I can do is point out some of the tributaries.

If you peruse the References section or do an online search, you'll find the concept of "low-fidelity" prototyping popping up circa 1990 from authors like Jakob Nielsen, Bob Virzi, and Tom Tullis, to name just a few. A few people in high-tech companies were using the technique during the 1980s and earlier (Chapter 2 has some interesting examples courtesy of Robin Kinkead). As far as I can tell, the method was around for a decade or two before it showed up on the radar screen of the average person (such as yours truly) involved in product development.

Paper prototyping has an estranged cousin: participatory design. I say estranged because many authors discuss one or the other without mentioning (or perhaps even realizing) how related they are. It's probably most accurate to describe paper prototyping as a subset of participatory design. There is a whole body of literature pertaining to participatory design, which has been around for decades; this is where the river starts branching off into more sources than I can do justice to.\*

And of course, prototyping in its general sense has been around for a long time in engineering disciplines and life in general. Before chiseling the first wagon wheel out of stone, I'm willing to bet that primitive humans prototyped and tested the concept using some other medium. So any work pertaining to prototyping could be considered a precursor to this book.

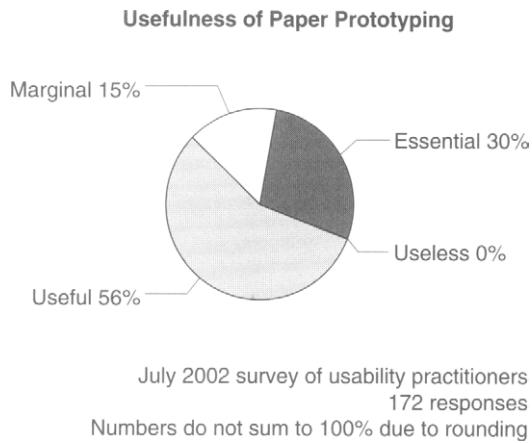
## Usefulness of Paper Prototyping

In July 2002 I conducted an online survey of some usability professionals about their use of paper prototyping and their attitudes toward it. Figure 1.12 shows that of the 172 people (most of them usability specialists) who responded to the question, "What is the importance of paper prototyping to your work?" 86% answered either "Essential" or "Useful." ("Useless" was one of the options, but no one chose it.) Although this survey isn't very scientific (it's a biased sample because I issued the invitation on usability-related discussion lists), it's still a pretty good indication that paper prototyping is no longer a fringe technique.

But this book is not about the history of paper prototyping, or even the present. It's about the future. Specifically, *your* future in using it to develop better products.

---

\*If you're interested, anything by Michael Muller is a good place to start. He's done considerable work in participatory design. Think of him as your guide to that branch of the river.



**Figure 1.12** Answers to the question, “What is the importance of paper prototyping to your work?” from a July 2002 survey of usability professionals.

## Audience for This Book

So let’s talk about you. You’re probably involved in the design, implementation, or support of user interfaces in some way. In other words, you might be any of the following:

- ◇ Software or Web site developer
- ◇ Project manager
- ◇ Interface/interaction designer
- ◇ Usability specialist
- ◇ Technical writer
- ◇ Graphic designer
- ◇ Information architect
- ◇ Marketing professional
- ◇ Quality assurance/test engineer
- ◇ Training specialist
- ◇ Technical/customer support representative
- ◇ Manager of any of these

Following are some assumptions I’ve made about your background.

## Programming

You don't need to have technical skills to create and test a paper prototype, although presumably (if there's any intention of implementing what you're prototyping) you'll be working with someone who does. Unlike software-based prototyping tools, for paper prototyping you need only those cutting, pasting, and drawing talents you perfected as a child. So although many of you might be programmers, I don't assume any technical knowledge about software development.

## Interface Design

Strictly speaking, you don't need to be an interface designer to create a paper prototype any more than you need to be an artist to glop paint on a canvas. Of course, to be a *good* artist you need some combination of talent, training, and practice. The same is true of interface design. Strictly speaking, this book won't teach you to be a good designer, but if you have the enthusiasm to learn, paper prototyping will give you a means to practice and thus refine whatever talent you may have.

## Usability Testing

It was hard to decide how much material to include about usability testing. Usability specialists who read this book certainly don't need me to tell them how to conduct tests. But for every usability specialist, there are probably 100 designers, developers, and writers who wouldn't mind knowing a little more about usability. There are already some good books about usability testing, and I don't want to reinvent the wheel. However, discussing paper prototyping without talking about usability testing is like trying to gossip without using pronouns.\*

My compromise is to give you a sense of how usability testing works, including some of the concepts and tips I found most helpful when I first started, but I leave most of the details to other books. I've concentrated most of the general usability

---

\*The next time you throw a party, ask your guests to sing "Let Me Call You Sweetheart" sans pronouns before you'll give them back their car keys—not because it's an effective sobriety test, but because it's funny.

testing information into a couple of chapters so that it's easy to skip for those who already know it.

## Usability for Everyone

For products to truly become usable, the development team can't rely on a handful of usability specialists to gather and interpret data from users; this creates a bottleneck that either slows the development process or forces usability activities to be skipped entirely (guess which). Thus, it's good for product teams to have a way to collect at least some of this usability information for themselves.

You wouldn't discourage a friend from taking singing lessons simply because he isn't an operatic tenor like Pavarotti. Similarly, I don't think it's necessary to be an HCI (human-computer interaction) guru to get started with paper prototyping. Paper prototyping and usability testing are common-sense techniques, and people in a variety of disciplines can benefit from using them. Yes, some will be better at it than others, but like my parents always told me, "You never know until you try."

On the other hand, usability specialists can spend decades developing their knowledge and skills. There is a great deal to know about the fields that comprise HCI, including cognitive psychology, social psychology, statistics, experimental design, data analysis, interface design principles, and probably others. I don't mean to summarily dismiss all that collective wisdom by implying that any idiot can conduct a good usability study. My philosophy is simply that anyone who cares about making better interfaces can benefit from learning the basics of paper prototyping and usability testing. But there is always more to learn.

## Author Background

It's probably already clear that I am a practitioner, not an academic. I gained much of my experience in hard-to-use interfaces because I spent the first decade of my career *creating* them, first as a software engineer and then as a project manager. (The building controls company where I worked was full of well-meaning people, but this was back in the 1980s, when most people, including me, knew nothing

about usability.) Although my software skills have fallen by the wayside, I still have some clear memories of the pressures and challenges involved in product development. I remain sympathetic toward development teams, especially knowing that technology certainly hasn't gotten any simpler.

In addition to my computer science background, I also (for reasons still not entirely clear to me) picked up an MBA along the way, which gave me a tentative grasp of business concepts.\* Working at a small consulting firm (User Interface Engineering) for 6 years gave me a very practical, whack-upside-the-head grounding in sales, marketing, and finance. Like the difference between profit and cash flow—profit looks nice on paper, but cash flow is where your paycheck comes from. So I approach usability from a business perspective, trying first to understand the importance of the product to the company and to its users, then looking for ways to identify and manage the risks. After all, if an otherwise good company makes too many mistakes and goes out of business, everyone loses. Paper prototyping fits perfectly into that mind-set because it lets you make (and fix) most of your mistakes before the product goes out the door.

I am not a researcher or scientist, and I don't even consider myself an expert in HCI; it's a big field, and I'm sitting in a small corner of it. There are many good papers and articles about paper prototyping, but for the most part I've confined the references to their own section at the end of the book. I mention some of the especially interesting ones along the way, however. Although this book discusses many topics for which there is HCI research, this book isn't about the research—it's about what real product teams do. Scientific studies are a fine thing; the experience of practitioners is equally important but less often published. So I've made an effort to include many real-life examples, anecdotes, quotes, and case studies from my clients and peers—material you won't find anywhere else, because although many of these people know enough about paper prototyping to write their own books, they don't have the time.

## Terminology Used in This Book

Terminology is a perennial challenge in any high-tech field; technology evolves faster than the language used to describe it. Beneath the surface of seemingly

---

\* And also confused me because I no longer was sure which Dilbert character to identify with; sometimes I feel sympathy for the pointy-haired boss.

innocuous words such as *designer* lurks a myriad of meanings waiting to confuse or even alienate people who've latched onto conflicting ones. I certainly don't want to redefine the English language, but I should explain the conventions I've adopted in this book. In most cases I'm using a definition that may be broader than what you're used to, so keep that in mind if you take exception to how I've used a particular word. I defined earlier what I mean by paper prototyping, so here are the rest:

**Computer, computer.** In paper prototyping, a human being simulates the behavior of a computer. To differentiate the silicon-based entities from the carbon-based ones, I'll use *computer* when I'm talking about a machine and *Computer* to refer to a person who's simulating a computer.

**Product, interface.** I use the terms *product* and *interface* somewhat interchangeably—you might be developing shrink-wrapped software, an intranet site, or a handheld device. Although *product* implies something that is sold, maybe your interface isn't. I use both these terms in an inclusive sense that encompasses not just the screens but also help, manuals, training, hardware, packaging, and even tech support. If the user sees it or interacts with it, I count it as part of the interface.

**Product team, development team.** Using similar reasoning, *product team* means anyone who works on any of the aforementioned, including tech writers, marketing, training, and customer support. In other words, anyone who has a direct or indirect effect on the end user's experience of the product is part of the product team. Sometimes I use the term *development team*, but I'm not referring only to those who, as a programmer friend of mine puts it, "rearrange 1s and 0s all day." If you're making/testing/supporting something that users will eventually come into contact with, you're part of the development team.

**Release.** *Release* refers to that moment when real users obtain the official version of the interface and start using it. *Release* is commonly used for software, but there are synonymous terms for other types of interfaces—*launch* for a Web site, *deployment* or *rollout* for an interface with internal users.

**Screen.** Although the literal definition of *screen* refers to a computer monitor, I use it in a generic sense to mean any piece of the user interface, be it a Web page, LCD display, form, dialog box, and so on. When I talk about "the screens," I'm referring to whatever it is that you're prototyping.

**Test versus Study.** Usability *test* refers to one session with a user. A usability *study* is a series of usability tests conducted over one to several days.

## Of Interest . . . Comparing Apples to Apples

In any discussion of *paper prototyping* or *low-fidelity prototyping*, it's important to agree on what is meant by those terms. People use these terms to mean different things. If someone is using a different definition than yours, you may completely misinterpret each other's conclusions about the technique. It's a good idea to keep this in mind as you do further reading on the topic of paper prototyping.

As an example, consider the article "Low vs. High-Fidelity Prototyping Debate" by Rudd, Stern, and Isensee (1996). This article thoughtfully outlines the pros and cons of high- and low-fidelity prototyping. However, the authors' definition of low-fidelity prototyping is different than the one used in this

book. They assume that the paper screens will be *shown* to users rather than having users *interact* with them in the context of completing tasks. This is a very important difference, so it's no wonder that some of their conclusions about paper prototyping appear to contradict mine. If you read the article carefully, though, you'll realize that the different definitions are at the heart of the apparent discrepancies. Once you account for the terminology differences, most points of disagreement vanish.

Whenever I hear someone start talking about paper prototyping, I've learned to ask, "Please tell me what you mean by *paper prototyping*." This often prevents me from misinterpreting what they're talking about.

## Chapter Overview

In several years of teaching paper prototyping, I've found that the most natural learning sequence doesn't necessarily follow the way you'd do the activities in real life. For instance, it's easier to understand some of the strengths and weaknesses of paper after you've made and mock-tested a prototype, or lacking that experience, at least seeing plenty of examples.

This book is organized into four parts. Depending on your background and interests, it may make sense for you to read the chapters in a different order than how they appear in the book.

## Part I—Introduction

The next three chapters provide an introduction to the what, why, and how of paper prototyping: what paper prototyping is, why it's useful, and how to prototype various interface widgets.

Chapter 2, Case Studies: Several detailed examples of usability studies conducted with paper prototypes are included so that you can see the sorts of things that companies learn from them.

Chapter 3, Thinking about Prototyping: A lot of reasons, both practical and psychological, are provided to help explain why paper prototyping is a useful technique. In addition, some of its drawbacks are discussed, although the details are reserved for Chapters 12 and 13.

Chapter 4, Creating a Paper Prototype: The widget-level view of paper prototypes and how to include interaction, help/documentation, and hardware are discussed. This chapter is presented out of its proper order (logically, it's part of Chapter 7) because some people find it useful to see many examples when first learning about paper prototyping.

## **Part II—Process: Conducting a Usability Study with a Paper Prototype**

This is the practical, how-to part of the book. It assumes you've decided to try paper prototyping, so it describes how to go about it. (Those who are still undecided might want to read Part III first and then come back to this section.)

Chapter 5, Planning a Usability Study with a Paper Prototype: This chapter includes an action plan of the activities, people, and schedule you'll need when you're ready to try paper prototyping. (If you're already familiar with how to conduct a usability study, you may want to skim this chapter.)

Chapter 6, Task Design: Good usability tasks are vital in paper prototype testing. This chapter explains why and how to create good tasks.

Chapter 7, Building the Prototype: The process of creating a paper prototype around a set of usability tasks and holding internal walkthroughs to prepare for usability testing is described in this chapter. (Refer to Chapter 4 for details about how to prototype specific interface widgets.)

Chapter 8, Introduction to Usability Test Facilitation: For those who haven't previously conducted usability tests, this chapter provides enough guidance to get started. Testing with two users at a time (a.k.a. co-discovery) is also discussed. Experienced facilitators can probably skip this chapter.

Chapter 9, Usability Testing with a Paper Prototype: Everything from how you explain paper prototyping to what you do when a user tries to do something you haven't prototyped is covered.

Chapter 10, Observers: Observers are an integral part of usability testing. This chapter explores the benefits and risks of having observers in the test room and how to ensure that they will behave appropriately.

Chapter 11, Data: Capturing, Prioritizing, and Communicating: This chapter describes how to take good notes and prioritize the data you've collected and also discusses ideas for capturing and communicating what you've learned from your prototypes.

## **Part III—Deciding to Use Paper**

The following three chapters delve into the nitty-gritty of what paper prototyping is and isn't good for, whether it might make sense to use for your project, and the sorts of objections you might face from your co-workers when you pitch the idea to them.

Chapter 12, What Paper Is Good For: Paper prototyping is a useful technique, but it is not perfect. This chapter outlines the kinds of usability problems that paper will (and won't) typically find.

Chapter 13, The Politics of Paper: It's one thing to convince yourself that paper is worth trying, but it's quite another to convince your co-workers. Common concerns about paper prototyping (validity, bias, professionalism, and resources) are discussed.

Chapter 14, When to Use Paper: Various circumstances of your project, staff, and development environment can affect your decision of whether to test with a paper prototype. (Strictly speaking, this chapter should precede Chapter 5, which also pertains to planning. I've placed it here because it may contain more detail than some people need.)

## **Part IV—Broadening the Focus**

This section encourages you to think about how paper prototyping relates to the overall process of product development.

Chapter 15, Examples of User-Centered Design: Descriptions are provided of some real companies that use paper prototyping as one of several techniques in their user-centered design process.

Chapter 16, Final Thoughts: This is a short but thought-provoking list of my own unanswered questions about paper prototyping.

And last of all, there's a References section with all the books and papers referenced throughout the book, plus plenty of suggestions for further reading.

## No Bad Examples!

There are many examples of interfaces in this book, but there are no bad examples! That's because all the examples I use are intended to illustrate techniques, not designs. In other words, I'm not critiquing anything. The point of the examples isn't to illustrate "good" or "bad" design but rather to give you ideas and insights about the *process* of improving your own design.

## Companion Web Site: [www.paperprototyping.com](http://www.paperprototyping.com)



There's a companion Web site for this book at [www.paperprototyping.com](http://www.paperprototyping.com). There you'll find downloadable versions of materials shown in this book, including worksheets, handouts, and forms. The site also contains links to places where you can purchase the supplies used in paper prototyping and links to papers and articles available online. The symbol in the margin is used throughout this book to indicate the presence of corresponding materials on the Web site.