Chapter 2

Case Studies

So what is paper prototyping good for? In a nutshell, it lets you create and refine an interface based on user feedback *before* implementing it. This chapter contains several real-world examples where paper prototypes provided that important—and often surprising—feedback.

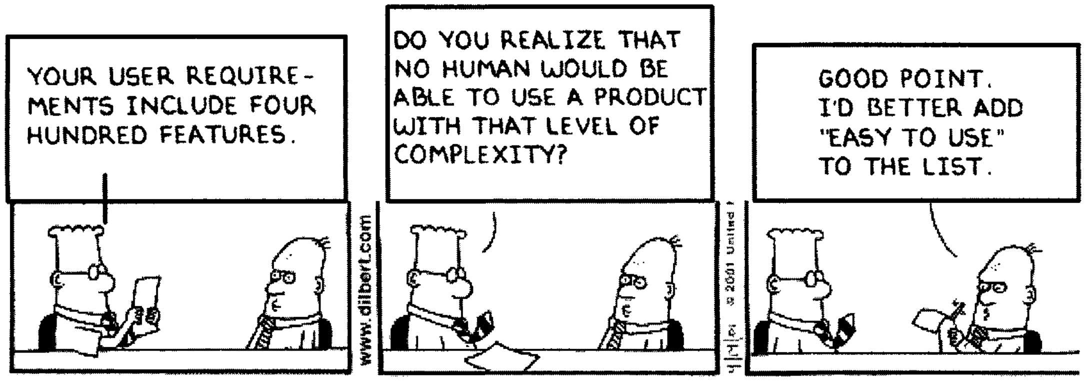
To illustrate that paper prototyping works for a variety of interfaces, I've chosen a software application, a Web site, a Web application, a telephone display, and a small touch screen. In each example, the product team created a paper prototype of the interface before a working version was available and tested it with about a half dozen users.

You'll notice several types of findings from these case studies:

1. **Usability issues.** This broad category contains all the sorts of things you'd probably expect to find from usability testing—confusing concepts, poor terminology, lack of feedback, layout problems, improperly used widgets, or any other situation in which the users can't get the interface to work the way they need it to. But usability issues are just the beginning.
2. **Missing (or misspecified) functional requirements.** It's common for users to have some needs that the product team isn't aware of at the start of the project, or the team may have a mistaken assumption about what functionality will satisfy a user requirement.
3. **Preference for one design alternative.** Sometimes there are different ways to provide a function and it's a coin toss to developers which one is easier to implement (but instead of tossing that coin, they debate it ad nauseam in meetings). Users, however, may have a firm preference for one or the other, which will show up pretty readily in usability testing.

4. **Priorities.** No company has unlimited resources, so it's important to separate the gotta-haves from the nice-to-haves. Occasionally a paper prototype will reveal functionality that isn't as important as the team had thought—a feature that draws a ho-hum response from users can be moved lower on the priority list or even dropped, thus saving work.
5. **Issues outside the user interface.** Interestingly, paper prototypes can reveal some issues that one might think of as being outside the scope of the user interface, for example, the credibility of the company or the implications of using the interface in a social setting. When the interface and tasks are realistic enough, test participants start extrapolating to their own environment and thus they can anticipate problems. (Caveat: Not all the problems—some problems can be found only in real-life use and won't show up in a usability lab, even if you test the released product.) These issues can result in changes being made in the interface or elsewhere, such as marketing or training materials.

You'll also see that there's quite a variety in the scope of issues revealed by these case studies—everything from high-level strategic issues ("Are we building the right thing? Will the market accept this product?") to low-level and specific ("Does this control offer the right degree of magnification?"). In my experience, it's common for paper prototype tests to turn up both high-level and low-level issues, so these examples are pretty typical.


YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.
www.dilbert.com
© 2001 United

DILBERT © UFS. Reprinted by permission.

Software: The MathWorks

Contributions by Jennifer Lymneos, Mary Beth Rettger

This paper prototype from The MathWorks is of the Control Point Selection Tool (cpselect), which is part of the Image Processing Toolbox 3.0 (Figure 2.1). This application is used by scientists to align images (for example, from geographic surveys) to detect differences between them. The user works with an overview and detail view of each image, choosing comparison points at the same location on each image.

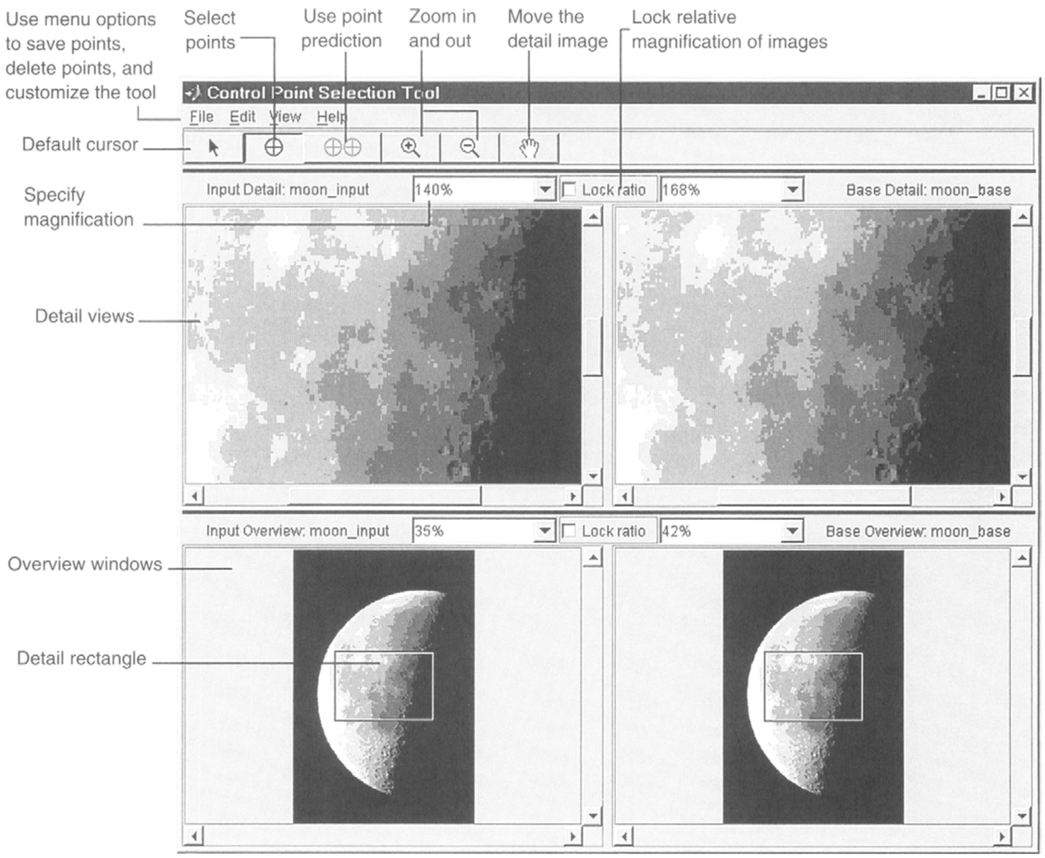


Figure 2.1 As shown by this screen from its documentation, the Control Point Selection Tool lets the user examine two images, including an overview and detail area for each. Clicking on any image adds a control point to the appropriate location in the other images.

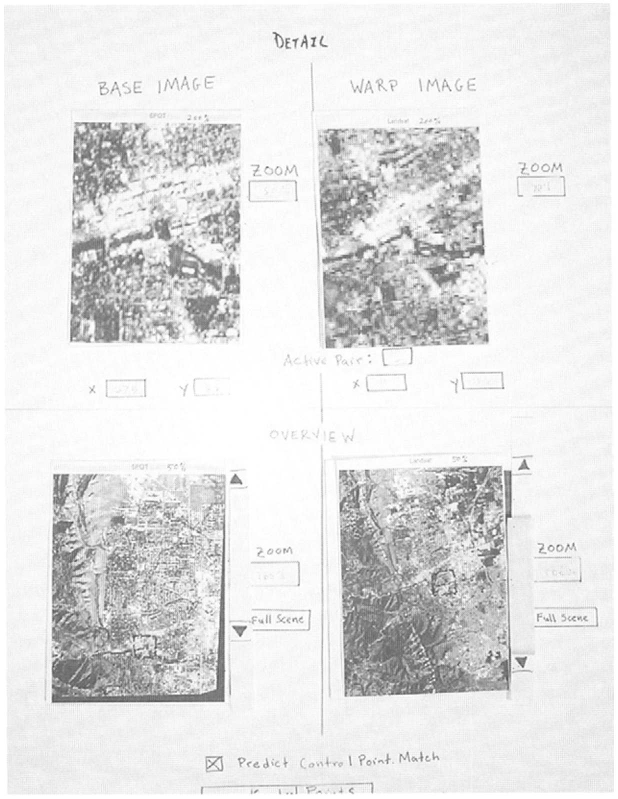


Figure 2.2 One of the paper prototypes used in testing. Not shown here are the colored dots on pieces of transparency that were placed on top of the images to indicate the control points.

In 2000 The MathWorks was working on the first release of cpselect. They had done their homework and knew a fair amount about their user population and requirements, but the developers were having a tough time getting started. In a scientific application such as this there are many technical details, and the team was getting caught up in them before they even had consensus on the high-level approach to the interface. So they created a paper prototype (Figure 2.2) and conducted several rounds of usability testing, refining the prototype as they went.

The Team Had Questions . . .

In the early prototypes, the team had a number of fundamental questions about what users needed and which design variations would work for them:

- Did the overall information display work? Did the users really need to see both a detail and an overview for each of the two images, or would just an overview do the job?
- When users dug down into the details of an image, what were they looking for? How did they want the control points to work? Initially, the team thought of control points as having two states: manually selected and predicted (that is, placed as a result of an algorithm using data from previous manual point selections).
- Did users want to specify the degree of magnification, or would a simple "zoom by a factor of 2" control suffice?
- How important was it to make the interface consistent with similar tools that their users had worked with?

. . . And Paper Prototyping Answered Them

The team learned that the four views were in fact needed and that users wanted a high degree of control over the magnification—zooming by a factor of 2 each time is conceptually simple, but it doesn't cut it when you're looking at large land masses. The team also learned that their "Fit to Window" control, which some had argued was too vague, didn't trouble users, because when they needed to manipulate the magnification they'd use the more precise tools.

Some interesting subtleties emerged in regard to the control points. The team realized that there were really several states—and combinations of states—that the points could have. For example, a point had different meanings depending on whether it had a corresponding match in the other image or was awaiting a match and whether the match was predicted or manually selected. Although users liked having the predicted points, they needed to distinguish them from manually selected points because they often had to tweak the position of the predicted point manually. Once the team understood the set of point states that users needed to see, they created a symbol for each one and a legend to help the users learn what the symbols meant.

One happy finding was that it was okay for cpselect to differ from other tools that users had worked with. The product team had worried about this because the market for technical computer software is highly sophisticated and some of The MathWorks' customers had written their own software applications to help with image processing. If customers were accustomed to a different way of doing things, they might not like cpselect. Fortunately, because cpselect incorporated the features users needed but with a simpler interface compared with the home-grown tools, its differences proved to be benefits rather than drawbacks.

Customers Are Happy

All in all, the team conducted five rounds of paper prototyping, refining the interface after each. (Although the "five rounds" may sound arduous, each round included only two or three users, and the team went from a blank slate to a ready-to-implement design in about 3 months, while working on other projects at the same time.) The resulting interface was much different than what they'd started with, and the product team believed it was also much better. Although the paper prototype didn't answer every question they had about the interface, there was a much smaller (and highly focused) set of issues to be solved later in the development process when a working version was available.

The Control Point Selection Tool shipped in April 2001, and customer feedback indicates that the tool is easy to learn and helps them do their image registration work. Their biggest request is not for bug fixes but for new tools to automate even more of the process. Jennifer Lymneos, a Usability Specialist who worked on cpselect, adds, "By the end of paper prototype testing, months before the tool was released, we were confident that the design was a good one. None of the feedback from users has contradicted what we learned from the paper prototypes, nor has it been related to fundamental aspects of the tool. In fact, many of the enhancement requests are small things like where the legend should go and what its title should be. For the first release of a brand-new tool, I think that's a pretty good indicator that we got the basics down right."

Web Application: Centra Symposium

Contributions by Ronnie Thomson, Drew Wolff

In the summer of 1996, developers at a start-up company called Centra were working on the first release of their flagship product Symposium, an Internet-based training environment. Symposium is a Web application that offers a live, "virtual classroom" environment where students with microphone-equipped PCs can hear and talk to the instructor and other students while viewing class materials on their computer screens. Distance learning was a new concept at the time,

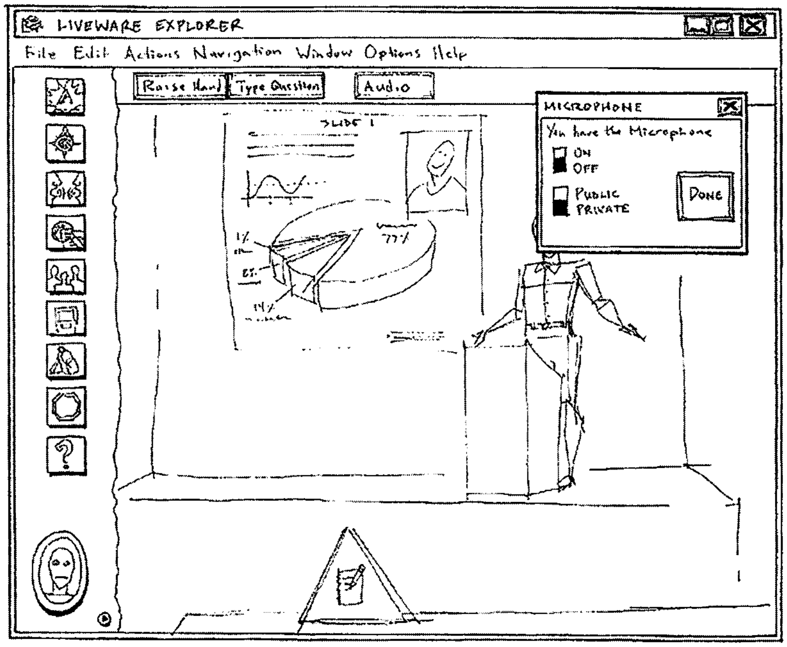


Figure 2.3 The initial paper prototypes of Symposium (called Liveware at the time) featured avatars for the instructor and students. In this example, the instructor is giving a PowerPoint presentation to the class.

and Centra hoped their product would appeal to large corporations with geographically dispersed employees, who would use Symposium from their office or home to receive training without having to travel.

The initial design was ambitious (Figure 2.3). The designers wanted to create a virtual classroom in 3D. Using avatars, students would navigate various rooms (such as a library or classroom) and talk with the instructor and each other via an audio channel. The avatars offered the potential for richer communication through facial expressions and gestures, such as hand-raising to ask a question or head-scratching to indicate confusion.

The product team had questions about whether the navigation worked, whether students could access the online course material, and how well they could participate in the classroom activities. Usability tests provided answers to these questions, and we also found a couple of surprises:

- ◇ Rather than enhancing the students' learning experience, the 3D virtual reality interfered with it.

- ◇ The product had some important social issues: Users wanted to know who could see and hear them, and they raised concerns about using the interface in a professional work setting.

A Return to Flatland

The paper prototype had stick-figure drawings of the avatars and sketches of the classroom environment. A total of eight users (two at a time) participated in co-discovery usability tests in which they were asked to register for and then take an "online" class. Although it lacked a 3D appearance, the paper prototype sufficed to show us that a 3D environment would interfere with the students' learning experience.

We discovered all manner of confusion related to the avatar, starting with the fact that some people had never heard the term before. Questions arose about the appearance of one's avatar, and users spent time fiddling with the customization controls. The 3D navigation proved cumbersome, with users voicing objections such as, "Why do I have to walk my guy into the classroom? Why can't I just take the class?" It also wasn't clear what viewpoint the avatar should show. Should the interface appear as though the user was looking through the avatar's eyes, or should the user have an outside perspective where they could see their own avatar along with everyone else's? How would the classroom "seating" be handled? (With a spatial layout, students in the back of the classroom wouldn't be visible to those in front, and students in the front might block others' view.) None of these questions had easy answers, and they all distracted users from the learning experience.

The bottom line was that the cutting-edge 3D functionality got in the way of the real purpose of the application—enabling large companies to provide training to geographically dispersed employees. The designers realized that they were jeopardizing their entire business model with an interface that felt too much like a video game.

Based on the feedback, Centra dropped the 3D functionality, which shortened their development schedule. Ronnie Thomson, Director of Engineering, explains, "Time to market was our major concern, and the 3D functionality was our biggest challenge from a technical perspective. Our original plan was to implement it in C++. When we dropped the 3D, we were able to use Java instead, which was a simpler development environment. The end result was that we released rev 1.0 sooner than if we'd stayed with the 3D and C++."

Social Considerations

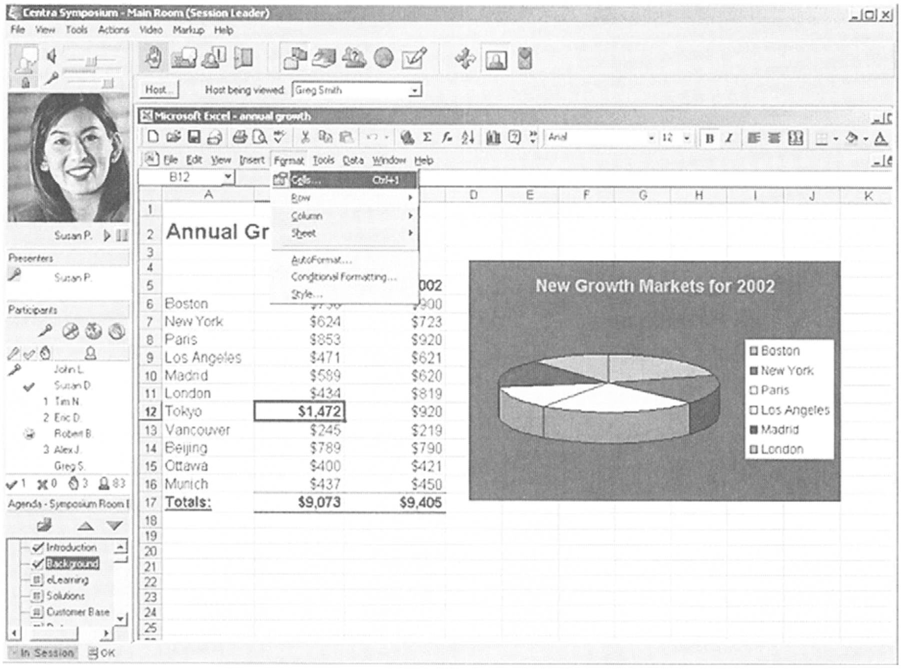
The paper prototype tests were conducted with everyone—users, facilitator, Computer, and observers—sitting in the same room, so obviously the users realized that people in the test setting could hear them. But the users were able to extrapolate the test setting to real life and identify some of the social concerns that they would have if they were to use this interface alone at their desks.

Because of bandwidth constraints, it wasn't possible to support a full-audio environment. The instructor had a microphone and could "pass" a second microphone to one student at a time. In testing the prototype, we found that the test participants wanted a clear indication of when their microphone was on. (In a similar paper prototype test of a video-conferencing system, one of my colleagues found a situation in which a test participant mistakenly believed that the person on the other end of the pretend system couldn't hear him. He made a joking remark that would have been embarrassing had it happened in real life.) In both of these cases, the designers modified the interface to make the microphone indication more salient.

I learned the live microphone lesson the hard way. During a break from teaching a large class, I went to the ladies' room without turning off my cordless mic! I really wish someone would have found this problem with a paper prototype and implemented a warning buzzer that goes off once the mic moves more than a certain distance from its receiver.

When the 3D functionality was removed, the designers realized they still had to give students a way to see who else was in the class. They added what they called the "people panel"—a listing of the class by name, including visual indication of which person has the microphone (Figure 2.4).

Users also raised some interesting concerns about using the interface in a work setting. With the 3D version, people worried that their co-workers might think they were playing a video game instead of working. They also anticipated the possibility of interruptions; for example, a person looking at a computer screen and wearing headphones might appear to co-workers to be listening to music rather than taking a class. Centra eventually offered kits for students, including a "do not disturb" sign, so that co-workers would know that the person was engaged in training and thus not interruptible.


Centra Symposium - Main Room (Session Leader)
File View Tools Actions Video Markup Help
Host...
Host being viewed Greg Smith
Microsoft Excel - annual growth
File Edit View Insert Format Tools Data Window Help
B12
Cells... Ctrl+1
Row
Column
Sheet
AutoFormat...
Conditional Formatting...
Style...
New Growth Markets for 2002
Boston
New York
Paris
Los Angeles
Madrid
London
Susan P.
Presenters
Susan P.
Participants
John L.
Susan D.
Tim N.
Eric D.
Robert B.
Alex J.
Greg S.
Agenda - Symposium Room I
Introduction
Background
eLearning
Solutions
Customer Base
In Session
OK

	A	B	C
1			
2	Annual Gr		
3			
4			
5			002
6	Boston	[illegible]	[illegible]
7	New York	$624	$723
8	Paris	$853	$920
9	Los Angeles	$471	$621
10	Madrid	$589	$620
11	London	$434	$819
12	Tokyo	$1,472	$920
13	Vancouver	$245	$219
14	Beijing	$789	$790
15	Ottawa	$400	$421
16	Munich	$437	$450
17	Totals:	$9,073	$9,405

Figure 2.4 This screen shot from Symposium 5.3 shows how the instructor avatar was replaced by a live video feed and student avatars by the Participants panel. Note the microphone symbol to show which class member is speaking.

Six Years Later . . .

As of 2002 Symposium is still alive and well in only two dimensions. Ronnie Thomson, Director of Engineering, reports, "Interface changes in subsequent releases have been relatively minor—most of our focus has been on providing enhancements to the functionality such as the ability to share applications. We tried making radical interface changes in rev 5 (without paper prototyping them first), but customers hated it so we went back to what we had. We've also used the same basic UI model for other products." Drew Wolff, former Director of Product Marketing, concurs: "The target market includes a lot of people who are learning about computers and applications specific to their company. Having to learn how to use Symposium would defeat its purpose. Fortunately, the interface is elegant and intuitive—no need for a book that explains how to use it—so people can concentrate on the class material."

E-commerce Web Site: Priceline.com

In the mid 1990s Walker Digital, a small Connecticut think tank, was preparing to launch a new online service. Their name isn't exactly a household word, but the Web site they launched became one: Priceline.com. Back then, the site was several months away from its initial launch, and they were still trying to figure out exactly how this unique* method of selling airline tickets would work. There were some pretty significant risks, both with the technology and with people's willingness to do business in this new manner. The product team wanted to mitigate the risks by finding out whether average people understood this service and were willing to use it to purchase airline tickets.

For those who aren't familiar with how Priceline works, you tell them when and where you want to fly (U.S.-originating flights only) and how much you're willing to bid for an airline ticket between those cities on those dates. You also give them your credit card number. Priceline checks with several major airlines and lets you know if any of them is willing to sell you the ticket you want for the price you named. If so, bang—you've bought a ticket (usually a nonrefundable one), which is charged to your credit card. The site targets travelers who have flexibility in their plans; the customer can't select the airline or flight time but may be able to get a price that's lower than advertised fares.

Initially, the product team had been focusing on the technical challenges (which were considerable) and also on selling the concept to the major airlines. But they recognized that without consumer acceptance, their clever new business model for selling airline tickets wouldn't get off the ground, so they decided to conduct some usability tests. I worked with several members of the product team to create a paper prototype of the site. All of the pages were drawn by hand because the site was still its conceptual stages. People had some ideas about how the interface might work, but just about everything was still open for debate.

Two members of the product team participated in 2 days of test sessions by "playing computer." We recruited seven people who fit the profile of Priceline's target market and asked them to work with the paper prototype to perform various ticket-purchasing scenarios. We found several important issues, including three showstoppers:

*So unique, in fact, that it's patented. (Patent Number 5897620: Method and apparatus for the sale of airline-specified flight tickets. Issued April 29, 1999.)

- Asking for an email address to enter the site met with complete resistance.
- The site (including the company behind it) had to establish its credibility before people were willing to transact.
- Three days was too long to wait for an answer.

After the first two usability tests (four users total), we'd already observed so many serious issues that three of us spent a few hours that evening completely revising the prototype. The next day's tests confirmed that most of our changes were improvements. Equally important, we found some things that the team *didn't* need to worry about:

- People didn't need to see probabilities of getting a ticket; they were going to do their own homework on pricing.
- People understood the concept of submitting a binding offer. They had some questions about it, but overall they understood what the site was offering and how it might benefit them.

Following are the details of what we learned from the paper prototype tests.

Email Address

Some of the developers had already predicted that users wouldn't be willing to provide personal information up front, but the team was still arguing about it. Some of the Marketing folks insisted on asking people for their name and email address before letting them into the site. (Don't laugh. Remember, this was 1997, when the word *e-commerce* wasn't even in the dictionary* and before it was commonly understood that people would rather have a root canal than give out their email address for marketing purposes.) The feedback was unanimous. All seven users said they would either leave the site or enter bogus information. This evidence was enough to convince the proponents of aggressive data collection that it was a Bad Idea, and it was dropped.

Binding Offer, Trust, and Credibility

One of the big questions for Priceline was whether people would understand the concept of the binding offer—that by submitting a credit card they were committing to purchasing a ticket if Priceline found a suitable one that an airline was willing to sell.

The results were interesting. There were indeed problems, but not in the way we expected. On the plus side, the test participants did understand the nature of the binding offer and indicated that they'd be willing to make purchases this way if the deal was attractive enough. However, giving their credit card to an unknown Web site was quite another matter. As one user said, "Anyone can put up a Web site. How do I know these guys are legitimate?"

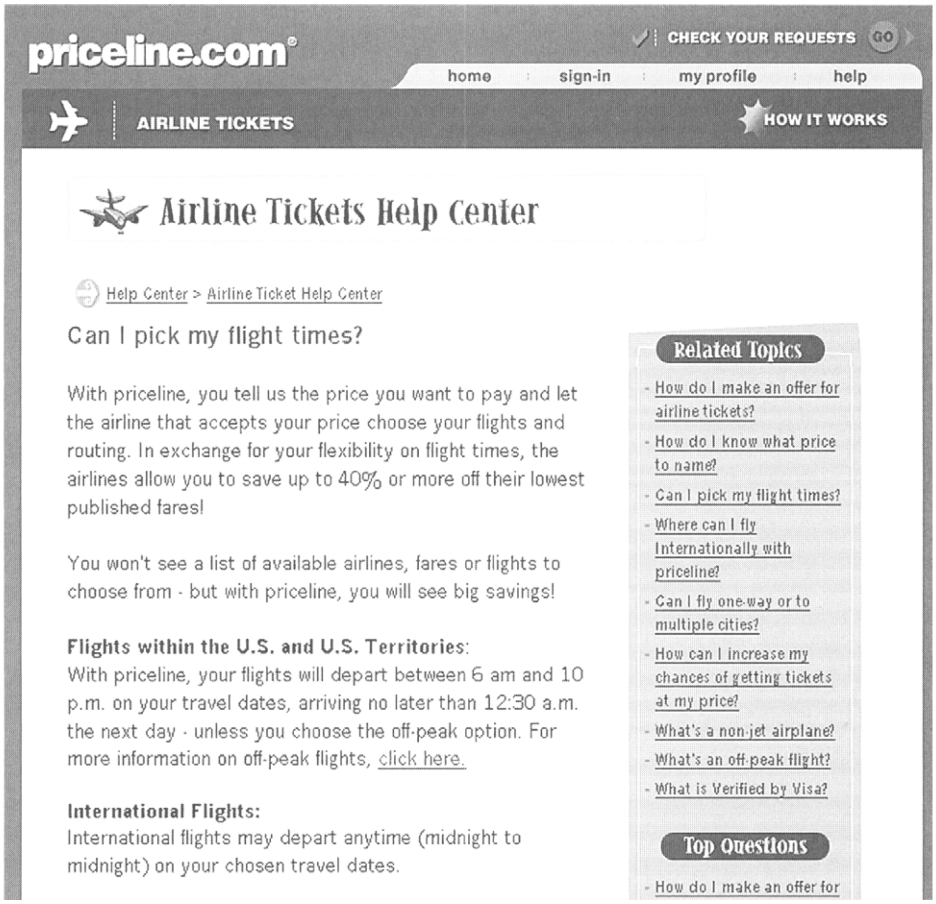
So we had to think of ways in which an unknown Web site (and company) could convey its credibility. In the testing, the users told us that seeing real flight information would help convince them that this site had a legitimate relationship with the airlines. Unfortunately, users sometimes want things they can't have. Priceline wasn't able to show a list of flights, but they realized they needed to do *something*. For example, even the relatively simple idea of including some company information, such as their office address, the names of company officers, and a toll-free number, helped reassure users that this was a real company (that they could then research further using conventional methods such as calling the Better Business Bureau, which a couple of users said they'd do). And in their help topics, Priceline explicitly says that they don't show flights, which may not be ideal to users but at least prevents them from looking for information that doesn't exist.

Ultimately, Priceline also launched an extensive advertising campaign featuring actor William Shatner that helped establish their brand in the minds of American consumers. I'd be very surprised if they still had the "Are these guys legitimate?" problem today (although they might face it all over again if they expanded internationally).

3-Day Response—Not!

There were significant technical challenges in matching the user's offer against the airlines' databases to see which ones would accept. The developers weren't sure how long it would take to query all the databases and respond to the user. In the paper prototype, we told users that they'd have an answer within 3 days. We quickly learned that this was unacceptable. If people couldn't know right away, they wouldn't use Priceline.

This wasn't good news to the product team because this requirement was much stricter than they'd hoped, and now they had a harder problem to solve. Fortunately, they were able to beef up their technology and the site now promises an answer within 15 minutes. However, in the absence of data about what was acceptable to people, the site might have launched with the 3-day reply and failed



Airline Tickets Help Center

Help Center > Airline Ticket Help Center

Can I pick my flight times?

With priceline, you tell us the price you want to pay and let the airline that accepts your price choose your flights and routing. In exchange for your flexibility on flight times, the airlines allow you to save up to 40% or more off their lowest published fares!

You won't see a list of available airlines, fares or flights to choose from - but with priceline, you will see big savings!

Flights within the U.S. and U.S. Territories:
With priceline, your flights will depart between 6 am and 10 p.m. on your travel dates, arriving no later than 12:30 a.m. the next day - unless you choose the off-peak option. For more information on off-peak flights, click here.

International Flights:
International flights may depart anytime (midnight to midnight) on your chosen travel dates.

Related Topics

- How do I make an offer for airline tickets?
- How do I know what price to name?
- Can I pick my flight times?
- Where can I fly Internationally with priceline?
- Can I fly one-way or to multiple cities?
- How can I increase my chances of getting tickets at my price?
- What's a non-jet airplane?
- What's an off-peak flight?
- What is Verified by Visa?

Top Questions

- How do I make an offer for

Figure 2.5 This page from Priceline.com, April 2002, explains the trade-off between flexibility and ticket cost—a question that came up in paper prototype testing 5 years earlier, before the site had even launched. Also note the explicit mention that Priceline doesn't show flights, which at least prevents users from continuing to search for this information.

Probability and Pricing

Initially, the designers wanted to estimate the probability of getting a ticket based on the user's bid. For example, for a $300 bid, there's a 50% chance an airline will accept. A nice idea, but pretty darn difficult to do with any degree of accuracy. Fortunately, after hearing all the users describe the ways in which they currently searched for the best travel deals, it was clear that most people would do their own research on lowest available fares and base their offer on that. End of problem. The developers could safely ignore this challenge instead of solving it. (Last I checked, the site simply says, "Visit other travel websites to research the fares and itineraries available on your travel dates.")

Frequently Asked Questions

One of the Catch-22s of new Web sites is figuring out what the "frequently asked" questions are going to be for a site that hasn't launched yet. But this is exactly what we were able to do. A useful output of the paper prototype tests was the set of questions that people had about how Priceline worked. For example, "Is there a fee for this service?" (The answer was no, but the site didn't say so.) Whenever a user had a question, we'd give them an answer orally, then jot down both the question and the answer for future reference. At present, the site has a help section that addresses a number of questions (Figure 2.5), including some that arose during the paper prototype tests. Naturally, the content of the help section has evolved over the years along with the site and the Web-using population.

We conducted these usability tests of Priceline in early 1997, well in advance of their initial launch. Six years later, the site is still in business. Although I can't claim that those early paper prototype tests were solely responsible for the viability of the site, I'd like to think that they helped at least as much as William Shatner's singing in their TV commercials!

Small-Screen Display: Pingtel xpressa Phone Interface

Contributions by Hal Shubin, Rich Schaaf

Pingtel is the creator of xpressa, a high-tech telephone. The xpressa phone has a user interface—a small grayscale LCD display. In addition to the usual set of telephone buttons, the xpressa phone has 11 buttons around the display whose functions change depending on the context (Figure 2.6).

Pingtel was developing this phone and its user interface in 1999. The team had written specifications to document the flow of the screens but hadn't nailed down the interface yet. The designers wanted to make the phone interface intuitive, but in some cases they had what seemed to be equally viable ideas about how to implement functionality. During the design phase a number of questions arose, for example:

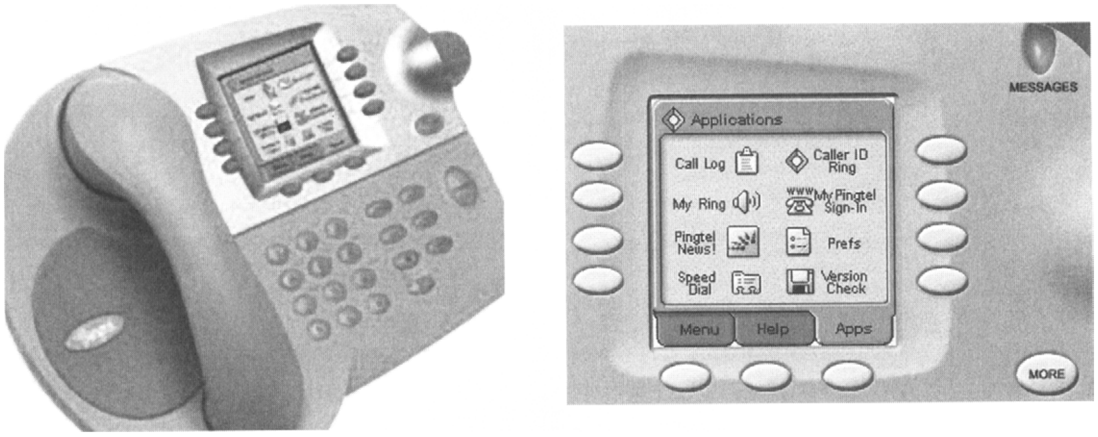


Figure 2.6 The production version of the xpressa phone and a close-up of its LCD display.

- Would users adopt an object-verb approach (for example, looking up a number first and then dialing it) or verb-object approach?
- Which was the easier method of searching for stored phone numbers—scrolling through the entire list or a divide-and-conquer method using the left-hand buttons to divide the list into successively smaller chunks? (See Figure 2.7.)
- Should they create novice and advanced modes, where only a subset of functionality would be available by default and the user would need to explicitly access the more advanced functions? Or would this just be confusing?

The right answers weren't obvious, and the product team didn't want to waste time implementing the wrong approach. They could have mocked up a version of the interface in software, but this would have taken some time. Besides, they believed it was important to see how users interacted with the physical buttons. So they opted to create and usability test a paper prototype of the xpressa interface. Because of the small display and the need to get the spacing right, they created the screens in PhotoShop using a format created by a graphic designer and used overlays for each individual screen.

The team conducted four tests of their paper prototype. Users were asked to complete various tasks (such as calling a previously stored number or transferring a call) by touching the paper prototype buttons and to explain their actions aloud as they went. One of the team members played Computer, swapping screens in

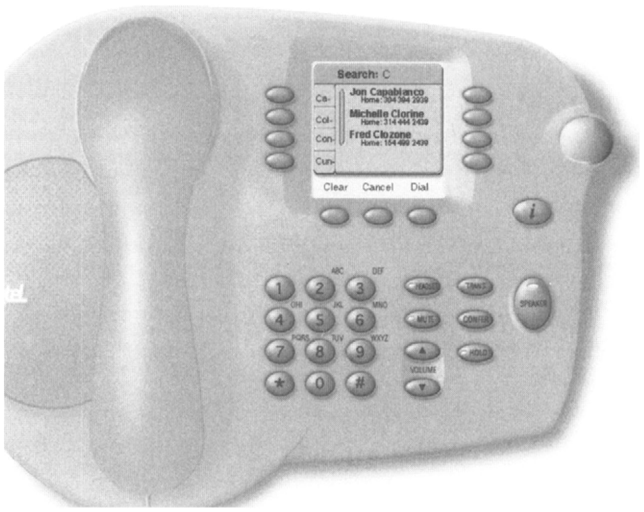


Figure 2.7 One of the paper prototype screens used in testing the search functionality, where users were asked to look up Mary Connolly's home number. The function of the 11 physical buttons around the display depends on context; in this example the four right-hand buttons are not used.

response to the users' actions. There was also a physical model of the phone sitting on the desk so that users could see what the real phone might look like, but they interacted only with the paper prototype.

Findings

The team got answers to their original questions and more. Users naturally used the object-verb method, so there was no need to support the verb-object approach. The divide-and-conquer method for searching was better than scrolling, although eventually the team came up with a type-in search method that was superior to both. The novice mode didn't work. Although hiding advanced options seemed like a good thing, it was too hard for users to find them when they were needed—they had no idea where to look.

The team also found out some things that they hadn't expected. In looking up names in the phone book, none of the users thought to use cell-phone style text entry, where you have to press the "2" key three times to get the letter C, etc. Users naturally pressed one key for each letter; for example, to enter *Smith,* they typed 76484 instead of 77776444844. This is the approach used by audio phone directories, and people tried this approach without being told that it would work this way. (The team had known about both models beforehand; what was surprising was how clear the preference was for the second approach.)

Hal Shubin, the usability consultant who worked on the design and testing of the xpressa interface for Pingtel, summed it up: "Paper prototyping removed a lot of the extraneous baggage; things the developers tended to focus on (shapes, colors, etc.) that weren't relevant to testing the interaction model. The paper prototype helped us learn a lot quickly, without writing any software." For more information on how the xpressa phone interface was prototyped and tested, see the Interaction Design Web site at *www.user.com.*

Touch Screen Interface: Jukebox Car Radio

Contributions by Samantha Lizak, Kristin Grasso

This example is a little different because it comes from a university course rather than from industry, and thus wasn't ever implemented. But this paper prototype is interesting because it simulated a touch screen that incorporated texture. Carnegie Mellon University students Samantha (Sam) Lizak and Kristin Grasso prototyped a jukebox-style music system for a car as part of a class project for a Human Factors class taught by Professor Sara Kiesler. The project's corporate sponsor had provided functional specifications (such as support for up to 1000 music albums), and the class worked in small teams to design and test a prototype interface.

The Modal Touch Screen

Because space is at a premium on an automobile dashboard, Sam and Kristin used a modal design. Depending on which of three physical buttons was pressed, the 5- × 6-inch touch screen console was used for the music system, climate controls, or other functions. For example, when in music mode, the left- and right-pointing arrows skipped over songs, and the up and down arrows controlled volume. In temperature mode, the arrows controlled hot/cold and fan speed, and the six rectangles on the bottom matched the standard airflow choices (feet, feet and face, defrost, and so forth). (See Figure 2.8.)

Before making the paper prototype, Sam and Kristin reviewed touch screen layout guidelines to ensure that their design followed the recommendations for

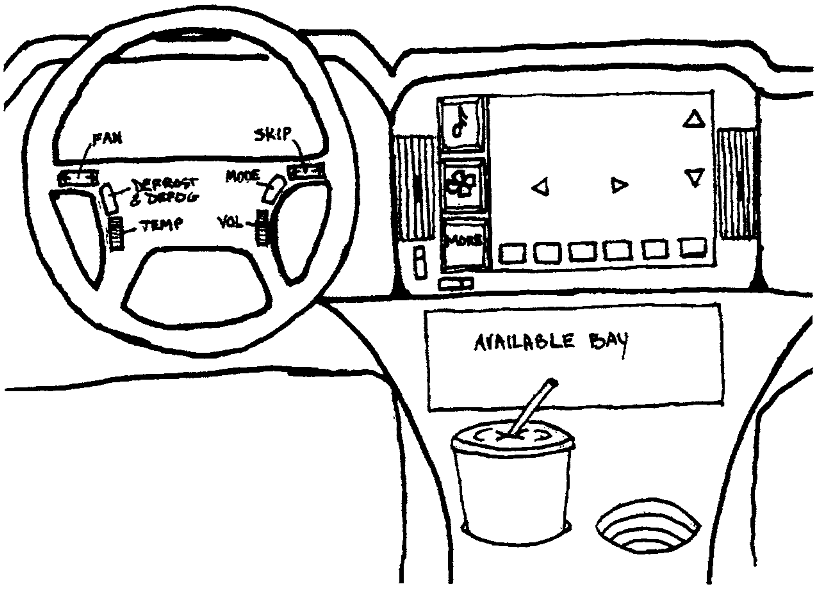


Figure 2.8 The conceptual sketch for the car music system interface. The music note, propeller blade, and "more" buttons are physical buttons that control the major mode of the display. The touch screen console is the mostly empty area above "available bay," and the triangles and rectangles represent textured buttons. (The console isn't normally blank. It displays the appropriate screen depending on the mode.)

button size and separation. (Kristin notes, "You discover how small 5 × 6 inches really is when you're trying to cram a lot of information onto the screen!") To get the spacing right, they made a template of the screen layout, copied it, and then drew each individual screen by hand.

Adding Texture

One drawback of a touch screen is that the user must look at it because the buttons are graphical rather than physical. Sam and Kristin first got the idea of incorporating texture into their interface after doing some contextual inquiry sessions (they rode around with people in their cars, taking notes about how they operated their car stereos). They observed that most people reached blindly for the radio controls and used touch to guide them to the correct one. The students believed that judicious use of textured buttons would help users operate the controls while keeping their eyes on the road.

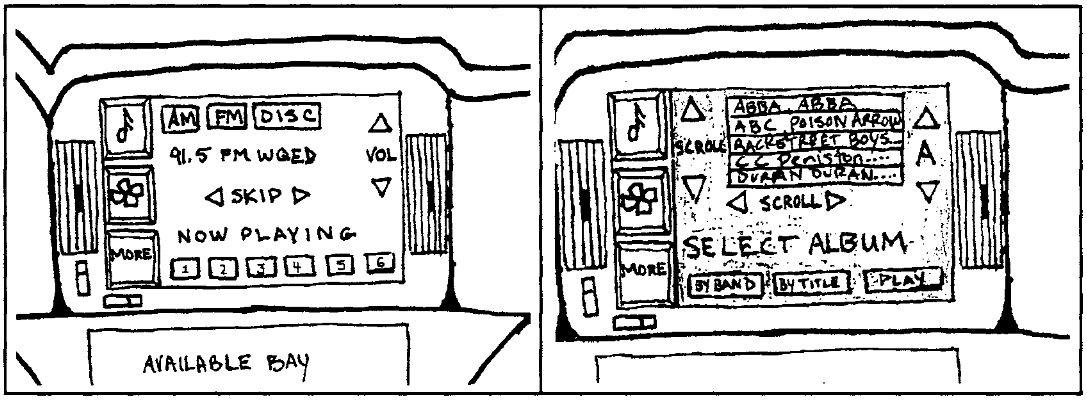


Figure 2.9 These two prototype screens illustrate different uses of the display area and textured buttons. The left image shows the radio display, where the six buttons were used in a conventional manner for preset stations. The right image shows the jukebox, where the three bottom virtual buttons each include two textured buttons.

The textured control buttons consisted of six rectangles along the bottom edge, left- and right-pointing triangles near the center about an inch apart, and up- and down-pointing triangles along the right edge. To add texture, the students used a clear plastic overlay and damaged the plastic over the buttons by poking it with a pen. In some modes, there were two textured rectangles per touch screen button. For example, on the screen shown in Figure 2.9, the user would feel two textured buttons for the "By Band" control, but both would do the same thing. This flexible mapping of physical to virtual buttons allowed the virtual buttons to be larger on screens where fewer buttons were needed.

Successful Interface, Successful Learning Experience

The students conducted five usability tests of their prototype. They placed each screen inside a clear plastic sleeve like those used for overhead transparencies. They put all the screens into a three-ring binder so that whichever of them was playing Computer could flip to the next screen based on user actions. The user sat at a PC-based driving simulator, and one student held the prototype off to the side, about where it would be in a car.

The students actually did their first usability test in a car driven slowly around a parking lot. Professor Kiesler decided that was a little too realistic and stipulated that for safety reasons, prototypes were to be tested without driving.

Sam and Kristin observed that people did use the textured buttons as they had hoped. The users were able to keep their attention primarily on driving and still operate the controls correctly. Even with only five users, there was a consensus that users wanted some additional functionality, such as a "profile" feature to store preferences for sound levels, temperature, and radio presets. The students also learned that some features they had considered (such as playlists) were either not important to users or were not practical without a keypad, which would be inappropriate to use while driving. The feedback from their paper prototype quickly revealed which of their ideas were workable and which were not.

Because this paper prototype was created in academia rather than in industry, its success is measured not only by how well the interface worked but also by how much the students learned from it. In addition to the basics of touch screen design, Sam and Kristin gained experience in balancing design constraints. The small display size forced them to keep simplifying the functionality and interface, and the paper prototype let them experiment with ways to make the most common functions the easiest to use.

Historical Examples of Paper Prototyping

Contributions by Robin Kinkead

The idea of testing interface mock-ups has been around for decades. Long before the graphical user interface (GUI) was born, engineers and industrial designers were testing prototypes of airplane cockpits, medical equipment, cash registers, and so on. Robin Kinkead, who has worked in the human factors field since 1965, was involved in a number of pre-GUI prototyping projects. Following are two of them from the early 1970s.

1971, NCR: Postal Mail Sorting Machine

In the early 1970s, much of postal mail sorting was still done by hand, and NCR was developing a machine to partially automate the process. In machine-assisted mail sorting, a postal worker would read the zip code from each envelope and key it in so that the machine could route it appropriately. One important design question was whether envelopes should be presented in a vertical or horizontal sequence. To work as quickly as possible, operators would look at the next envelope as they typed the zip code from the current one. Robin needed data on whether it was faster to move one's eyes left to the next envelope or up to it. He and his colleagues hypothesized that the horizontal sequence might be faster because the side-to-side eye movements required were similar to reading.

To find out, Robin built a paper prototype using foam board and stiff paper. He shifted envelopes manually while the user pressed buttons to indicate which way arrows went on the simulated addresses. The answer? Contrary to what they expected, vertical was faster. It was a shorter distance, and addresses are more compressed vertically.

1974, Xerox: Stenographic Translator

The Xerox Stenographic Translator was a color-screen editing station for computer-translated stenographic (court reporter) notes. The development team at Xerox knew that the user interface was going to be complex, that users had to navigate among several screens, and that representations of particular screens would change as well while the user edited text. Back then, they had no way at all of making a software prototype of the concepts (this was pre-GUI), so paper drawings of screen states were the logical choice.

The prototype consisted of 250 pages of screen drawings (naturally, similar screens were created with the aid of a Xerox copier). Stenographers were instructed on how the device was going to work and then were asked to perform several tasks, including bringing up notes, editing them, adjusting the steno's "profile," and finally printing. Depending on the user's choice, the experimenter flipped to the correct response page.

This technique proved extremely successful. A complete, complex user interface in an almost unknown medium was designed, tested with eight people, and refined in just 4 weeks. Stenographers using the 12 beta test machines were able to quickly learn and use them for creating written transcriptions of courtroom proceedings.* Unfortunately, Xerox never took the device beyond the beta phase, although Robin did receive a patent for the user interface.

Summary

One of the fascinating things about usability tests, whether or not they use paper prototypes, is how often product teams are surprised by what they learn. The team usually has many questions going into the usability study—and maybe even some accurate predictions about where the problems will lie—but the real value in usability testing lies in its ability to surprise the product team with answers to questions they never would have thought to ask.

As the saying goes, it's not the rattlesnake you see that will bite you. It's inherently difficult to know where your own blind spots are—usability testing helps you find them. Many of the problems described in these case studies would have had a detrimental effect on the success of the product, and in some cases the entire company, had they not been found before release.

As powerful as usability testing is, a paper prototype leverages that power one step further by letting you gain all that knowledge even earlier in the project, before implementation. Although all the problems described in this chapter could have been found by testing a working version of the interface, the product teams were glad they didn't have to wait until then.

* Robin told me: "The beta test units worked so well that Larry Tesler, then at Xerox PARC, said after he tried it: 'That interface was *really* friendly!'" Robin believes this may have been the first time anyone ever described a UI as "friendly," soon to become the much overused "user-friendly."