Chapter 15

Examples of User-Centered Design

Let's look at some examples of how people use prototyping as just one part of the larger picture of user-centered design. I asked three usability specialists to describe how they have used paper prototyping and other user-centered techniques at their company or clients. You'll notice that the examples have some things in common:

- **Early feedback.** These companies recognize the value of figuring out what's needed *before* building it, so they seek feedback from users and internal stakeholders.
- **Several techniques.** Paper prototyping is an important technique, but not the only one. In particular, there are other user-centered activities that precede the first prototype.
- **Fast and iterative.** Designs go through multiple revisions. There's no expectation that the early designs will be perfect, only that they'll improve each time around.
- **Small and informal.** No big budgets, no huge research studies, no fancy reports. Just do it, learn from it, do it again if needed (and schedules allow), and then move on.

Example 1: The MathWorks

By Mary Beth Rettger

I'm Director of the Usability Group at The MathWorks, a developer of technical computing software. At The MathWorks, we use several user-centered design

techniques to help us understand what our users need and to ensure that what we build actually meets those needs. Here's how we applied contextual interviewing, paper prototyping, and usability testing to the development of MATLAB 6.

Contextual Interviewing

It's important to watch and/or interview customers in their own work environment, where they can easily show us examples of what they are trying to accomplish. This process is called contextual interviewing. Rather than just getting a laundry list of new features users want, they show us how they hope to use a new feature or explain what problems they are trying to solve with the current product. Armed with a thorough understanding of the problem, our software designers can come up with targeted solutions that address the users' needs.

During MATLAB 6 development we undertook a research project with the assistance of undergraduate students enrolled in an Engineering Psychology class at Tufts University. The project was a win-win situation—we obtained a lot of data with relatively little effort from within the company, and the students received course credit for their work. After some training in contextual interviewing techniques by our usability team, the students interviewed about 20 MATLAB users. The students wrote formal reports summarizing the "top 10" issues observed during the interviews. They also provided us with detailed interview results, including artifacts such as samples of users' work, outputs, and photos of the users' work environments. Finally, the students created several "Meet Our Users" posters that graphically illustrated several typical users and their work. (This is similar to the technique of creating *personas*, although personas are usually made-up examples of typical users instead of actual people.)

All this information was valuable to the development team. Developers who were eager to get a fresh perspective on the problems our users faced appreciated the top 10 lists and the user posters. The individual interview transcripts were even more important—developers pored through this information to get specific data with which to prioritize their efforts. Several key themes came out during these interviews, including requests that we improve MATLAB's Printing, Plot Editing, and help system. That's where the work artifacts came in handy: Seeing examples of real work that users were trying to do (or had difficulty doing) helped the developers gain a better understanding of where to focus their efforts.

Because the students were not MATLAB experts, we were still left with the task of determining which changes to make to MATLAB. Several development team members (including tech writers and quality engineers) went out on a handful of follow-up interviews with users, getting answers to questions that had arisen from

the student interviews. This combination of efforts helped us come up with an excellent list of priorities for MATLAB 6.

Usability Testing of Paper Prototypes

Once they understood what users wanted, developers were eager to begin coding new solutions, so we moved on to paper prototypes. Most often, the basic design was hand-sketched, and other user interface elements such as drop-down menus, dialog boxes, and error messages were created using sticky notes, tape, and glue.

Thanks to the contextual interviews, we had a good idea of what "realistic tasks" were for our users. Team members worked with one task at a time and walked through the process of completing it with the prototype. Because the developers were focused on the users' tasks, they had to think hard about the purpose and value of each feature. This ensured that only the essential parts of the interface were created—no need to add extra features if they weren't relevant to the users' work.

The MATLAB 6 release contained several tools, so it was practical to test each tool with only a small number of users. We selected customers who would typically use the feature we were testing. One or more developers acted as the Computer, responding nonverbally to the user's interactions with the interface. Each user was instructed to "think out loud" to help us understand what they were expecting from the product. Although users often gave us puzzled looks to start with, they quickly got involved and had fun with the game-like aspects of working with the prototype. More important, most users had no problem making the connection between the paper version and the end goal of a more usable product designed for them.

Members of the development team sat in the room and observed each usability test. Sometimes only one team member observed; other times there were half a dozen. After—or sometimes even during—each usability test, we made changes to the prototype. We did as many rounds as we needed to feel confident that we were on the right track (or in some less ideal cases, as many as we had time for), and the developers went on to code the functionality that had been hammered out on paper.

Usability Testing of Working Software

It was important for us to retest designs after they were implemented in code. In some cases, there were features that truly didn't test as well on paper as online.

Also, once implementation started, there were inevitable changes and compromises necessary to express the design in code. So we needed to validate the more finished designs.

Much of this testing was done in our usability lab: a two-room lab, with rooms separated by one-way glass. We generally had one developer and a usability specialist sit with the user in the testing room, and additional team members observed from the other side of the one-way glass. As with the paper prototypes, we gave users realistic tasks to complete, asked them to "think out loud" while they worked and to let us know if they experienced frustration or satisfaction with the feature being tested. The usability specialist or developer sometimes prompted the user for more information along the way.

All the team members were involved in observing the test sessions, although not every person observed every test. Team members took notes during test sessions on sticky pads, writing down one issue per sticky. At the end of all the test sessions, we held a debriefing meeting involving the whole team where we conducted an affinity diagramming exercise, grouping issues from these sticky notes into like categories. Once the categories were established, it was easier for the team to review the issues and decide what actions to take. Because the entire team was involved in learning what was working (or not) in the interface, everyone could contribute to the solution. Thus, the process not only improved the interface but also served as a useful team-building exercise.

Usability Nights

We've also had a lot of success with what we call "Usability Nights," which are a fast way to get informal usability feedback from a group of users. Basically, we take over a training room for an evening and bring in a teamful of developers and a bunch of local customers. Instead of having everyone work on a problem at the same time, we set up stations for each tool or interface, staffed by the development team member(s) who's working on it.

We usually ask the users to bring in examples from their jobs to work on. Typically we're only getting feedback on one or two interfaces, but on one occasion we tested several development efforts at once. Users rotated through each station, spending about 30 minutes at each. Depending on the stage of the project and what the developer wants to learn, they might have used a paper prototype, working software, or even a hybrid approach combining a paper prototype with the real software.

Usability Nights are a fast way to get a lot of information, and our customers enjoy them as well. They appreciate the opportunity to be involved in the development of the products they'll eventually use in their own work.

Example 2: IBM

By Thyra Rauch

At IBM, our human factors specialists use a variety of user-centered design techniques throughout the design and development process. Our push has been to move more and more to the front of the process, working early with users and iterating on early design solutions. Paper prototyping is one of those techniques used successfully by many groups in IBM at various points in the process. Here are some examples from some of my recent projects.

When I began work with Tivoli Systems (a company later bought by IBM) several years ago, we had to work very quickly on a novel project—a piece of software that our customers had no experience with. The problems we were trying to address could be solved in a number of ways, so first we wanted to get those thoughts down and run them by our customers.

Storyboards

Before we began prototyping, we often used storyboards to capture more of the scope and flow of the design proposal. Storyboards are much like paper prototypes, but typically broader in scope and not generally intended for input from the customers. We tend to do them on big sheets of paper that capture the narrative of the customer stories, questions that need to be asked, issues that need to be addressed, assumptions, user information, decision points, and the like. These documents become a record for the team, and the ideas that we have captured on them remain to be explored later as resources free up. We've gone back to storyboards as much as several years later because they are such a rich source of information.

Paper Prototypes

Following our storyboarding exercises where we explored the domain, we created some very rough paper sketches of several alternatives and went prepared, with a team comprised of human factors and visual design specialists, to do some "design on the fly" if necessary. The beauty of this approach is that we could do this kind of activity virtually anywhere: at our development lab, at the customer's site, or at one of their trade shows or conferences. In many cases, none of the alternatives we had sketched fully met the needs of the customers, so we did a lot of quick sketching, with the customers telling us what to draw. In some cases, the customers got

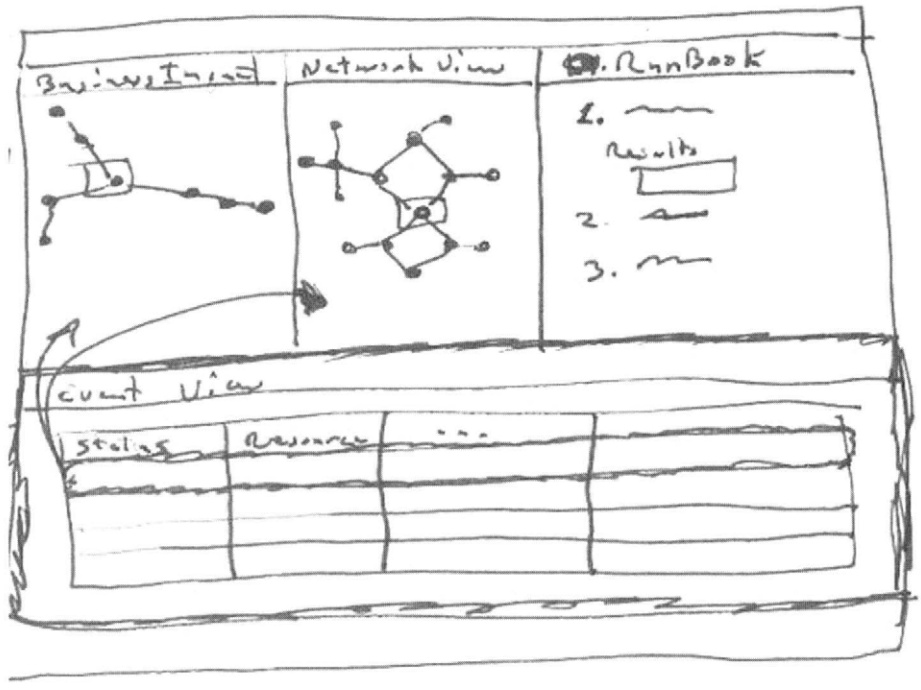


Figure 15.1 A very rough paper prototype screen that was used in usability testing.

so excited they grabbed the pens and started sketching themselves, which is part of the magic of this medium—anyone can participate.

Using rough forms such as the one shown in Figure 15.1, it's very easy to get people engaged in the process. On one hand, it's good to go in with a straw man because:

1. It shows you have done your homework in the domain.
2. It gives them something to react to, which seems to be easier them coming up with something from scratch.
3. It scopes the level you want them to work at by providing an example.

On the other hand, using paper prototypes encourages them to work with you to *evolve* the design, not just nod their heads in acceptance. The rougher the prototype, the better (to a point) because it readily conveys the idea that the design is open to input versus it's already been designed and is now cast in concrete with you merely getting their approval to proceed.

PowerPoint Presentations

Another technique we used a lot was to use paper screens printed off from PowerPoint slides. The slides included screen captures of existing product panels, screen captures where we did some cut-and-paste to make suggested changes, pencil sketches that were scanned in, and simple blocks with arrows for navigation and blobs where stuff goes. Again, the three main notions were being "open for input," "fast to produce and change," and "portable."

We've used a lot of PowerPoint prototypes in the past several years to communicate quickly and easily among team members and customers in remote locations because it's easy to distribute the prototypes online and then have everyone on a teleconference discussing them. At that point, the prototype really isn't "paper" anymore, but more in a class that we started terming "mid-fi" a number of years ago.

Mid-Fi Prototypes

At that point, I was part of a small multidisciplinary team that was working on one of the first Web applications. The product, BookManager BookServer, was an online document system. At that time, the notion of Web applications was pretty new, and we wanted to get the concepts of the system across to folks in as interactive a way as we could, as quickly as we could. We got bogged down trying to use low-fidelity or paper prototypes because they were not really interactive enough to show the power of our search engine, but we didn't want to spend the time to code high-fidelity prototypes, so the notion of medium-fidelity, or "mid-fi," prototypes using HTML was born. Our team, which included a programmer and an online help writer, could very rapidly mock things up for our customers, and even make changes on the fly. This project is described further in a paper I co-authored called "Web-Based Prototyping for User Sessions: Medium-Fidelity Prototyping" (Leone, Gillihan, & Rauch, 2000).

Summary

What all these methods have in common was that they allowed us to refine our understanding of users and what they needed. Storyboards captured our own preliminary ideas (and lots of questions) about how the interface needed to work, paper prototypes encouraged dialog with users, and mid-fi prototypes were useful for those questions that paper couldn't answer.

Example 3: Dictaphone

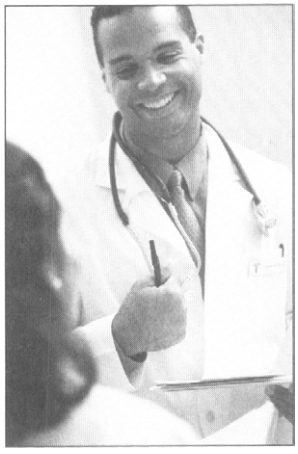
By Hal Shubin

I'm a usability consultant at Interaction Design *(www.user.com)*. One of my clients is Dictaphone Corporation, a provider of innovative voice technologies. Dictaphone is developing a handheld microphone used in conjunction with software called PowerScribe Workstation. Doctors use the microphone to dictate notes after each contact with a patient. The software converts the speech to text, which the physician can then edit via a combination of keyboard, voice commands, and buttons on the microphone. For example, the spoken phrase "Dictaphone: ankle sprain" alerts the speech recognition engine to put appropriate standard text in the dictation display. The physician can then navigate among fields to replace placeholders—indicate which foot was injured, what medication was prescribed, and so on. The microphone has functions like Dictate, Rewind, and Insert text, plus navigation controls used in conjunction with the software.

Developing hardware can be riskier than software because it's expensive to fix mistakes once the device has reached the manufacturing stage. The Dictaphone team used several rounds of prototyping, using progressively more realistic techniques. The methods allowed many internal stakeholders to participate in the design process, including members from several disciplines, such as software engineering, mechanical engineering, manufacturing, and a physician who acted as a proxy for the target user population. Here's a summary of what we did.

Usage Scenarios

In the earliest stages of design, I created some rough layouts and then Bob Hanson, the industrial designer, prepared scale drawings. Then we began working with a set of three usage scenarios we had created for a previous project—Dictaphone is proactive about understanding customers and their needs. Each scenario contained a fictional but realistic description of a doctor and a description of how he or she would use the PowerScribe Workstation. (An example is shown in Figure 15.2.) The scenarios helped us walk through several common tasks that we expected our target user population to do. Although this is not a substitute for usability testing, this scenario-and-walkthrough technique provided a context for our design discussions. We had one scenario that represented physicians who were not very comfortable using computers. This scenario came directly from the people we had met during usability tests of the PowerScribe Workstation software and was a good way to remind us to limit features and keep the design simple.



Goal	He's concerned about accuracy, both for the patient's sake and because of his dexterity problems.
Computer use	Average
Dictation	He dictates notes after each patient encounter, although he often reviews his dictation at the end of each day.
Affiliation	Family Health Care, Boulder, CO
Notes	Dr. Manos has difficulty using his hands due to a recent neurological problem. He gets a little support from the nursing staff during patient visits but can use a computer reasonably well. He represents some classes of handicapped users, but also people who simply have lessened dexterity.

Dr. Ramesh Manos
Pediatrics

Dr. Manos found the telephone-based dictation system easy to use. Because of his disability, he wore a headset instead of holding the handset and has a telephone with large buttons that are easier to press. The new PowerScribe Workstation set him back a little, until he worked out some accommodations. For example, when his hands tire, he holds the microphone with one hand and presses the buttons with the other. That's easier than trying to manipulate buttons with the hand he's holding it with. When possible, he uses voice commands for navigation . . .

Figure 15.2 Excerpt from a usage scenario for the PowerScribe Workstation. The scenario continues to discuss usage for a few more paragraphs, then has a table showing how Dr. Manos would use each proposed feature of the microphone.

Collaborative Design: Wood Blocks and Stickers

Bob and I wanted to show the development team our drawings, but first I wanted to have a group design session. Successful design is a collaborative effort—they bring us consultants in for our design knowledge, but the team members have the domain knowledge. An activity where people make something concrete is a great way to transfer that knowledge. A design exercise forces people to be explicit about their ideas, so you get better and richer information than when they just talk. We decided to do this exercise before showing them our drawings so that our ideas wouldn't influence them.

So we had them build microphone models. Bob prepared blocks of wood that were about the size of the microphone. We gave the team members pens and blank stickers in various sizes. We began the session with a discussion of the features they wanted on the device, just to get ideas going. As always, there was some skepticism—people are always a little leery when they have to do something that reminds them of art class! But once we got started, everyone really got involved. Some samples are shown in Figure 15.3.

At the end of the meeting, Bob and I showed them our schematics. Turns out that some of our drawings weren't valid any longer because our brainstorming discussion had simplified the design by moving some features from the microphone to the base it would sit in. So this exercise was very successful. It was hard work for them to turn their ideas into something tangible, but they realized that we had discussed the design in far more detail than before.

I think it's important to accept design advice from my clients because they've thought about the problem longer than I have. Team members had great ideas about arranging controls that helped Bob and me in our next round of design. Bob

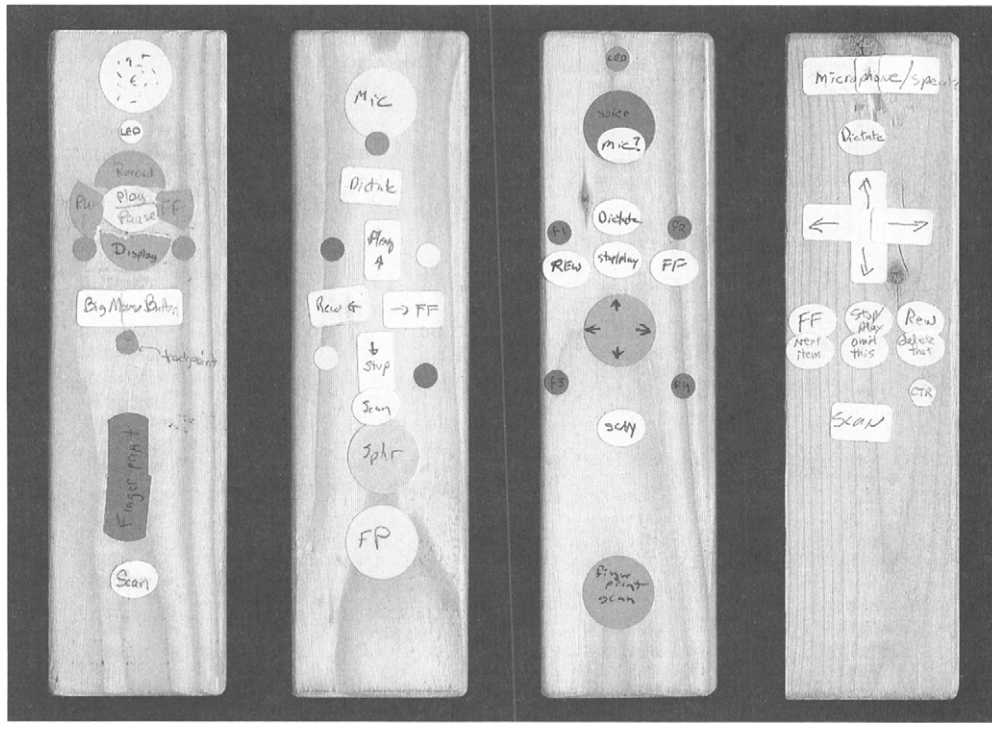


Figure 15.3 Using wood blocks and stickers, the team members experimented with button layouts and discussed their ideas. This exercise led to several changes in the assumptions about the design.

and I walked away with four complete prototype designs to use in our subsequent design work. It doesn't matter that we didn't develop all of the ideas—our job is to come up with a good design that makes the product successful.

Fome-Cor Mockups

We were concerned about the possibility of hand fatigue if users used the microphone for long stretches of time. The design schedule called for making full 3D foam models at this point, but they're expensive and take a week or two to make. We wanted some quicker feedback first. We had a pretty good idea of the relative importance of the buttons, and realized that it'd be okay for users to reach a little for the less-used ones. Our goal at this stage was to identify problems due to button location. Where would the user's hand rest? How hard would it be to reach the various buttons? Would it be okay to adjust one's grip to reach some buttons?

Bob made models of some button layouts using Fome-Cor (polystyrene foam sandwiched between two pieces of paper, available in various thicknesses). He drew the button layouts on Fome-Cor and cut them out to fit over the lower half of an existing microphone shell. Bob created about eight of these, representing five different configurations with three minor variations (such as a four-way rocker versus four separate buttons). Some examples are shown in Figure 15.4.

Here's where the usage scenarios came in handy again—by holding a model and pressing its buttons in accordance with anticipated usage, we literally got a feel for which designs were more comfortable to use even though the buttons themselves were only two-dimensional. We also got some insights about which buttons needed to be near other buttons. For example, our simulations showed us that we could increase efficiency by putting the button to initiate dictation near the one used to display the dictated text. At this stage, we also considered how complex the design looked, as well as some conventions such as having the Rewind button on the left and Fast Forward on the right. In a review meeting, we were able to rule out five of the eight models, leaving us with three.

Foam Models

Next we enlisted a model maker to build true 3D models so that we could further explore the look and feel of the microphone. We were able to settle on a single overall form, but the layout of the button arrays was still in flux. One change was to move the buttons so that they'd be easier to press when one picked up the micro-

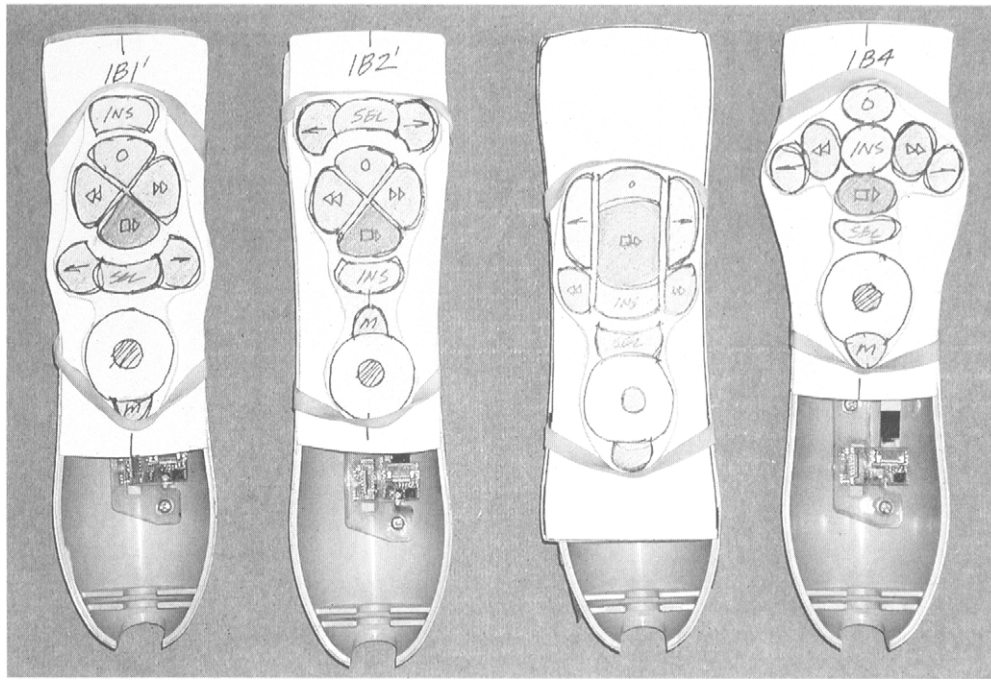


Figure 15.4 These prototypes (made of Fome-Cor, paper, and a microphone shell) enabled the development team to explore what it would be like to hold the microphone and use its buttons to perform various tasks.

phone. We also wanted to limit "regripping," although eventually we decided it was okay for secondary features.

The industrial designer and model maker then produced a final set of models, like the one shown in Figure 15.5. Bob came up with a very pleasing circular layout for the button array. He had the model maker create one microphone body with four replaceable alternatives for the button arrays. Some of the differences were concave versus convex button shapes and relative sizes of related buttons. I knew we were on the right track when the product manager picked up the model and a great smile came over his face! We also sent models to team members in other states, who gave us useful feedback about spacing and button sizes.

Looking Forward, Looking Back

As of this writing, the microphone is still under development. The mechanical and electrical engineers are working on the PC board and suggesting modifications that will make manufacturing easier. Bob is working with the mechanical engi-

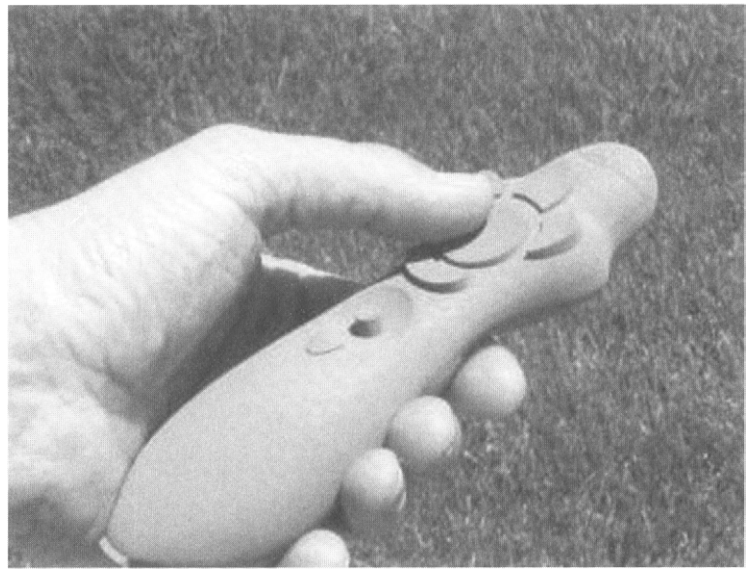


Figure 15.5 Foam models allowed the team to experiment with different button layouts.

neer to create a CAD database so that they can do the tooling for manufacturing. The model maker is creating a new button-array insert with greater tactile feedback between the buttons. I also plan to conduct usability tests, using the most realistic model that's available at the time.

Although all this prototyping may sound like a significant effort, any design mistake that made it into production would be very expensive due to manufacturing costs. David Pearah, Dictaphone's Senior Product Line Manager, Speech Products, sums it up:

> "The benefits of going through the prototyping steps are almost too obvious to mention. The design exercise with wood blocks gave us a good forum for discussing our ideas. The drawings, Fome-Cor models, and foam models each improved our understanding of the designs that Bob and Hal were creating.
>
> We are building something that physicians will hold all day long. Without a 3D model, there's no way to know how comfortable it will be. Designing the microphone was more complex than we thought it would be because curvature, button spacing, and tactile feedback all make a difference in the user experience. We needed multiple models to discover the subtle things. It's important to know when to stop making prototypes though; at some point you have to actually make the thing. It can never be 100% perfect. There are long cycles in hardware development and foam models are the cost of reducing risk. Although it increases time up front, it can save time later on. The process Hal describes worked well for us."