How to do Usability Testing[1]

Sarah A. Douglas
Computer & Information Science, University of Oregon

OVERVIEW

One area of concern common to developers of interactive systems is pinpointing where design fails for the first-time user and how to improve it. User-centered design often suggests performing usability tests. In this paper I will advocate a particular method of usability testing conducted with pairs of people who are continuously videotaped while performing selected typical tasks. The paper discusses the method in detail and its integration with standard software prototyping techniques and usability studies. It also discusses how to analyze a videotape. A detailed example, drawn from a three year project which developed a cardio-vascular simulation for use in teaching biology, illustrates the method. Finally, a description is given of informal confirmation of the learnability, practicality and universality of the method through several years of teaching it to programmers in user interface design classes.

INTRODUCTION

How do we create successful designs for interactive systems? User-centered design recommends the use of continuous usability testing of prototypes and implemented code before release as a commercial product. The general format of usability testing requires that (potential) users perform real tasks. The sessions are usually videotaped and then reviewed by the developers for human-computer interaction problems. The difficulty is that little information is available on how to effectively conduct a usability test and interpret the results.

The overall design process.

Usability testing should inform the user-centered design process from the earliest stages of development. Since user-centered design is an iterative model, usability testing should be conducted multiple times during the development process. From the user interface design standpoint, this

[1] This paper was created for the Intro to User Interfaces class. It abstracts parts of a book chapter, written by Douglas called Tutoring as Interaction, in Peter Goodyear (Ed.) *Teaching Knowledge and Intelligent Tutoring*, Ablex, 1991.

method insures that most serious usability problems will be flushed out by the end of development (Jeffries et al., 1991; 1992). From the programming standpoint, the rationale is that the greatest flexibility in design comes earliest when it is much easier to change things on whiteboards and mockups than in code. Designs can be implemented in paper storyboards and given to users for a very crude simulation of interaction. Although paper simulations allow earlier feedback on the success of the design, they cannot substitute for the complexity of true interactive computing. Consequently, it is very important to plan to schedule multiple usability tests with working software prototypes.

Rapid prototyping combined with high-level programming languages specially designed for user interfaces attempts to finesse this problem with automation support designed for flexibility and incompleteness of a design (Myers, 1995). The difficulty with a rapid prototyping tool is that it may not support all the interface actions desired for the final product, generate efficient code or generate code in a language that will interface to an application such as a data base manager. All prototyping methods require a highly modular implementation strategy for the software. Choosing that modularity may be critical to the success of the prototyping approach.

Once the prototype, whether paper mockup or running software, is created, a usability test is conducted with real users. Usability testing requires that participants perform several test problems or tasks in a scenario which recreates a realistic work or play environment. During the usability test, the participants' interaction with the software is recorded on videotape. Pre- and post-test questionnaires or interviews can be given to screen and or debrief them.

After the usability test is conducted, the design team reviews the video tapes selecting problem areas of interaction. After the team discovers all the problem areas and generates design solutions, these design changes are prioritized, trading off severity with complexity of programming. A new prototype is produced and tested again.

METHOD

In this section I describe the usability testing method in detail. We begin with preliminary planning and testing, and then describe the actual tests and analysis.

Preliminary Planning and Testing

No empirical study of human behavior can be done without alot of advance planning. Decisions must be made about what to test, equipment must be working, and people selected to participate. The following describes six critical steps in this planning process:

1. Identify purpose of test
2. Find a test site
3. Create participant materials
4. Prepare video equipment
5. Prepare schedule
6. Select users
7. Conduct Pilot test

Step1. Identify purposes of test

Before any detailed planning occurs, the developers and or the UI testing team must decide what they want to learn from the usability test. Many times their interest is in general usability: Where does the software fail?. At other times, the developers may be particularly interested in anticipated trouble spots in the design. For example, do users recognize the function of a particular icon? Finally, there may be testing of specific usability requirements that were generated during requirements analysis in early design. This usually occurs late in the development process when an almost finished product is anticipated. An example of a usability requirement is: "The user must be able to complete a simple query within 1 minute." This type of usability testing requires quantitative measurement of performance time which can be derived from the videotape.

It is critical that the testing have clearly described, written objectives. Likewise, it is necessary to have an understood method for obtaining the testing result from the testing process. Is it a qualitative analysis of the human interaction on the videotape, an average of participant task times, or a qualitative measure of responses in a pre- or post-questionnaire?

An example of a usability testing planning document is included in the Appendix at the end of this paper.

Step 2. Find a test site

It is important to find a quiet place to conduct the usability test where no interruptions can occur. The best setup is to have two rooms with the testing room containing the computer, video recording equipment, and any other necessary furniture such as chairs or tables. Either a one-way mirror or a video camera observing the participants can provide the usability testers with visual access. (More on this later.) If only one room is available, I believe that it is very critical that the developers not be present in the room where testing takes place. Participants and developers will often change their behavior to conform to the others' expectations. For example, if participants are having trouble with a particular task, developers can appear uneasy or even jump in and tell them what to do. In the most ideal of circumstances, the developers should have no contact with the participants at all, although it is beneficial if they watch the test from a remote site.

Step3. Create participant materials

The most important part of participant materials are the task scenarios. These consist of real tasks chosen relative to the purposes of usability testing. For initial testing, these are the core tasks (functions) that the system and user must perform. For a word processor these might be tasks such as opening a document and saving it, moving a paragraph to another location, or changing a word to bold. These tasks must be couched within realistic scenarios that are presented to the participants in written form. For example, "Imagine that you are to change the following document, shown with marked editing changes. The name of the file is stored on your hard disk as 'Letter to Mary'."

Other supplementary materials must often be prepared in addition to the task testing materials. Often participants might be give instructional materials, a summary of commands, or user manuals. These are themselves prototypes of the final product and don't have to be completely finished products. For example, the contents of on-line help may not be integrated into the system, but can be tested as a paper document. Again, the purpose of the testing determines what supplementary materials are to be given. If a simple walk-up and use system is being tested, or if there is a brief instructional demonstration before the actual test, then no instructional materials should be present. If, on the other hand, it is a CAD-CAM system with 500 functions, the user documentation may form a critical part of the system.

Both types of particpants documents (task scenarios and instructional or help documentation) should be neatly typed and clearly marked. Each task scenario should be on separate pages. If there are alot of pages, these materials can be put in binders for easy access and use during the testing phase. (An example of a task scenario is included in the Appendix.)

Finally, if either pre- or post-test questionnaires or interviews are used, they must be written out and tested. These materials assess the participants background and subjective experience. Pre-test questionnaires or interviews often ask the computer or software experience of the participant, their age and gender (if relevant). These are used to screen the potential participants or correlate the findings of the test with their background. Post-test questionnaires or interviews often ask the participant to rate the software on usability when compared to other similar products, describe in more detail the overall impression of the software, e.g. easy to remember the commands, useful help system, etc., or describe in more detail areas that were problems and suggestions on how to fix them. (Examples of a screening and post-test questionniare are included in the Appendix.)

Since usability testing requires either a paper or software prototype, it is critical to bring all the materials to the testing site. If it is a paper prototype, be sure all the pieces are available. If it is a software prototype, install it on the computer. TEST either type of prototype. Sometimes there are significant differences between the developers' software environment and that of the testing site. Finally, there may be software components of the task scenarios that must be installed. For example, a document file used for making text editing changes. Be sure there is table space for placing the materials next to the participants.

Step 4. Prepare video equipment

During the usability test, the participants' activities are recorded using both video, audio and, possibly, computer generated records such as keystrokes and pointing actions. Usually the videotape shows just the display, but sometimes a second video image showing the participants is added and integrated into the final tape. If a pair of participants is recorded, each can be separately recorded in audio on the stereo channels of the videotape. For later convenience in reviewing, the tape can be time-stamped on a frame by frame basis.

There are a wide range of videocameras and recorders available that operate in natural lighting. We have found a "camcorder" (portable camera with integrated recorder) on a wheeled tripod to be most useful. The camera is positioned in back of the participants, raised over their heads and aimed at the screen. If more than one image is integrated, separate cameras must be used, the images synchronized using gen-lock, and then processed in a Picture-In-Picture processor before feeding into the video-recorder.

Step 5. Prepare the schedule
Enough time must be given in the schedule to prepare the usability test materials and equipment, set it up in a testing site, conduct a pilot test, recruit and select participants, run the testing sessions, and analyze the data. The pilot (test) run of the usability test should be scheduled before the "official" tests with enough time to solve any problems encountered in the testing materials, software or equipment. In general, a usability testing session should last from 20 minutes to one hour. If it is longer than that, the participants tend to tire. If the testing must last longer than an hour, it is often possible to break it into multiple sessions. The scheduling of usability testing for all participants will depend on when the testing facility and testers are available, and when the participants are available.

Step 6. Select users
Usability testing always tests the software with typical users. The tester tries to match as closely as possible the participants with targeted users in terms of knowledge of the domain, work context and computer experience. Participants might be users with experience in the domain, or users of earlier versions. For the software under development, all these users should be "first-time" novices. The participants are NOT the developers, nor people pretending to be users of the targeted software. Participants can be recruited through many different means including newspapers, Web site, email or circulated posters. A description of the usability testing should be given, the approximate schedule, and any requirements as far as experience. If incentives, such as money or food, are offered they should be stated. (Note, incentives may often be necessary.) Always try to recruit more people than you think you might need, since people sometimes cancel out.

Either a single person or a pair of participants can be observed during a single session. Single participant recording is called *protocol analysis* (Lewis, 1982; Ericsson & Simon, 1984), and is the most common method. The participant is encouraged to "think aloud" while a solving problems or performing tasks. The primary goal of protocol analysis is to uncover the underlying mental life of the participant. Protocol analysis is useful in design studies because it provides a method for identifying "problems" users may have with the design, some elaboration as to why and perhaps ideas for improvement. Unfortunately this method has one major limitation familiar to psychologists. The demand to talk may cause participants to "make up stories". Also, it is highly improbable that reports of a person's own mental processes are scientifically valid (Nisbett, 1977). Despite these limitations, protocol analysis provides developers with two critical pieces of information: Where the design fails to achieve what developers expect, and some insight into why.

To eliminate the problem of making up stories, pairs of participants are used in a method called *constructive interaction* (Miyake, 1986; Suchman, 1987). It attempts to elicit verbal information within naturally occurring conversation. By using two participants a situation of collaborative problem solving is created whereby each participant must inform the other in an explicit verbal record about problems, causes, and solutions. During protocol analysis, the participant sometimes forget to think aloud and must be prompted. Likewise, during constructive interaction, the tester should try to select participant pairs who will feel comfortable with each other and collaborate. (Avoid using pairs of strangers where the pairs consist of different genders.)

Based on many years of experience with usability testing, we have come to use pairs of participants working together with three pairs for each cycle of usability testing. My reasoning is that three is the minimum number to differentiate universal problems from those which are more unique to individuals.

Finally, in some institutional environments (universities, research labs, etc.), there are formal criteria of experiments with human subjects. The institution will have policies and procedures explaining how to obtain review and approval. Even for occasions where there is no formal regulation, there are ethical issues with usability testing that should always be addressed, namely, informing the participant of rights, reimbursement, privacy, procedures and purpose of the testing. These topics are included in a consent form which is signed by the participant. Videotaping or working with strangers can often be distressing, and the participant rights include the ability to quit at any time. Participants should always be reassured at the time of the testing that the purpose of the test is to find problems with the software, not with them. Similarly, the videotape record should always remain private and the participant anonymous, unless the participant specifically allows it to be viewed publicly. (Such consent should always be in writing. An example is included in the Appendix to this paper.) Particular care should be taken in showing a videotape of a worker to the boss. It is easy to interpret the problems that are seen as lack of intelligence.

Step 7. Conduct pilot test

The pilot test is a test of what you have planned. The participants are chosen from the same testing population. However, the data is not usually analyzed since the purpose of the pilot is to verify that the equipment, both computer and videotaping is working, the software is working without any "application" bugs, the length of a session is reasonable, the participant understands the testing materials, and the tester knows what to do.

.

Conducting the Usability Test

Once the pilot has been completed and any changes made to the testing environment, the real usability testing can begin. The tester should arrive early and verify that all equipment and software is working, the written materials are available, and the environment set up. Have an area where participants can wait if they arrive early. Put up signs in the building informing them of the location of the test. When the participants arrive, greet them and have them fill out any preliminary documents, e.g. informed consent, or pre-test questionnaire. After turning on the video, the tester then begins the testing. (It is often useful to have a check list available for the tester so that a consistent set of instructions and activities are give to all participants. An example of such a checklist is included in the Appendix.) Often a usability test has an orientation or instructional period which requires that the tester show the participants basic preliminaries. This should be included on videotape. Once these are concluded, the tester should leave the room. This is very important since the participants will often keep asking the tester for help, a situation which would not be possible in the real world. A signaling system (buzzer, etc.) can be arranged for the participants to call the tester for help if they cannot continue.

During the entire testing period, the tester can take notes making observations about what is happening. If the tester observes a total collapse of the software, e.g., a system crash, he or she should intervene. (If video access or one-way mirror is not available, the tester should still leave the room, but will not be able to make observations or intervene during the testing.) If the testing extends beyond a reasonable amount of time, the tester should also terminate the test. An example of an observation sheet is included in the Appendix.

At the end of the session, the participants are often given a post-test questionnaire or interview to determine subjective assessment of the software, or to follow up on any unusual problems which occured. If it is an interview, it can be included on the videotape.

ANALYZING THE RESULTS OF A USABILITY TEST

Once usability testing is completed, developers are still left with the problem of systematic analysis of a primarily visual and verbal record. The videotapes are usually reviewed by the whole development team. This allows them to freely discuss possible problems, interpretations and alternative designs. During the review process a log is kept with the following general information:

- Name of the prototype
- name of participant(s)
- date and time of usability test
- date and time of analysis
- names of reviewers

For each problem observed on the videotape, the reviewers keep the following log:

- location on the tape in either frames or time
- the task the participant was attempting to do
- the stated intentions of the participant (Note this might not be identical to the task proposed by the testing!)
- the actions performed by the participant
- the participant's perceived effect from those actions
- analysis of the cause of the problem
- recommended solutions.

First, a pass is made through the entire tape, keeping a log and noting the frames (or times) where problems occur. The videotape is then reviewed a second time, skipping to the areas where problems had previously been observed. Discussion is then focused on causes and design solutions. For some events which cannot be immediately interpreted, a written transcript can be produced which is later analyzed. Such a transcript is coded with particular attention paid to verbal comments including pauses, pointing and gazing directed toward the computer monitor, computer input actions such as mouse pointing or keying, and any significant computer-generated user interface events. (A coding method for this type of transcription found in Luff et al., 1990 and will be show in a later example in this paper.)

After the team has reviewed all the taped sessions, discovered all the problem areas, and generated possible design solutions, a prioritized list of interface "bugs" is made which gives both the severity and the complexity of programming solutions. If all participants have problems with a particular part of the interface, it indicates that the problem will be common for most users. Severity notes that there is a range of problems from annoyance to catastrophic. Fixing problems also ranges from the trivial to a major redesign. A final decision must then be made as to which changes will be made to the prototype.

What do you look for in the video-tape and the transcript? Three primary aspects stand out. First, detect events where trouble occurred. Second, diagnose the cause. Third, propose design solutions.

Detecting trouble with the design as revealed by participants' behavior is fairly straightforward since the talk will be laced with expressions easily recognizable to developers: "Oh my! What happened?", "Wait, wait...", and "I don't understand this." Garden paths, i.e. users didn't know that or when things had gone wrong, when in fact they had, are infrequent but more difficult to detect. At a more structural level of analysis repetitions and restarts of a sequence of computer-based actions often signal trouble. In fact, it is often difficult to pinpoint exactly where the problem starts and a great deal of reviewing of the videotape is often necessary.

We have found that diagnosing the cause of trouble is often problematic for the developers and dependent upon the ability of the team to recognize users' intent and the failure of the system to respond as expected. At times intent is not verbally expressed and even when expressed, its intelligibility depends greatly upon the analysts' ability to place themselves in the context of the participants' work. This requires recreating rich visual and aural images from many sources, not just the already coded transcript. (Note that we do not code the whole videotape because of cost and time constraints.) Group analysis of intent frequently requires argumentation based on empirical evidence. A particular theory has to be convincing to the other members of the team. The evidence which discloses an underlying intent is often not found in the verbal record, but in a complex interweaving of a history of talk, computer monitor state changes and user input actions. Participants frequently point to areas on the screen and used indexicals ("this", "here"). Again, the full video-tape and not a transcript is critical in the analysis. Developers are often able to compare what they expect users' intent to be, that is the design rationale, with the actual intent. This underlines the domain knowledge necessary to really understand what is going on. Showing the videotape to persons not members of the design team and not HCI experts familiar with the domain, provide no aid in analysis.

Once the cause of trouble has been decided upon, e.g., no feedback, don't understand meaning of icon, no action availble for a function, etc., the developers should brainstorm possible alternative design changes. Solutions can often be found by observing the repairs the participants perform or by carefully analyzing the causes. It is critical that the developers think of the design in the large when proposing changes, because it is possible to make the situation worse. Iterative design assumes progress towards an improved product. However, in user interface design, this is often difficult.

AN EXTENDED EXAMPLE
Using usability testing to design the Cardio-Vascular Construction Kit

During one extensive software development project beginning in 1987 and extending over a three year period, I experimented with the use of usability testing in informing design. The opportunity presented itself with the development of a simulation program called the Cardio-Vascular Construction Kit (CVCK) to be used in teaching biology labs at the university level. In biology labs students often shared computers in a collaborative way which made Constructive Interaction very compatible with their actual context of use. The project also required us to develop workbooks for the biology lab exercises that would use the software and were intimately involved with the actual use of the software in a learning environment. This software is now available nationally as a software product in the BIOQUEST biology lab package available from the University of Maryland on CD-ROM and is thus not a "toy" example. This software has also had almost no complaints about the design of the interface or usability.

The design team consisted of myself as the primary developer, two professional programmers and one biology teacher. Our use of pairs of participants involved bringing two same gender, similar age and ethnicity, first-year biology students to our HCI laboratory. (We discovered that we would get much more cooperative discussion if we paid attention to potential social power asymmetries.) For each phase of the design we would typically bring in three pairs of participants. Very early in the design process, prior to software implementation, the participants worked with paper prototypes. Later, while using rapid prototyping tools, we worked with software implementations until the final program was completed.

The two participants were left alone in our lab and instructed to follow the instructions in a workbook. They were given a buzzer they could use to summon help. Our HCI laboratory was arranged to have three video outputs: one from a camera situated at a high elevation and acute angle behind the pair in such a way that we could see the CRT monitor and where they were pointing, a second camera at a side angle so that we could see the upper half of the participants' bodies with mutual gazes and talking, and a third direct NTSC output from the CRT monitor which was generated by the computer system running the software. (See Figure 1.) We also recorded from lavaliere microphones on each person to each stereo channel on the second camera output.

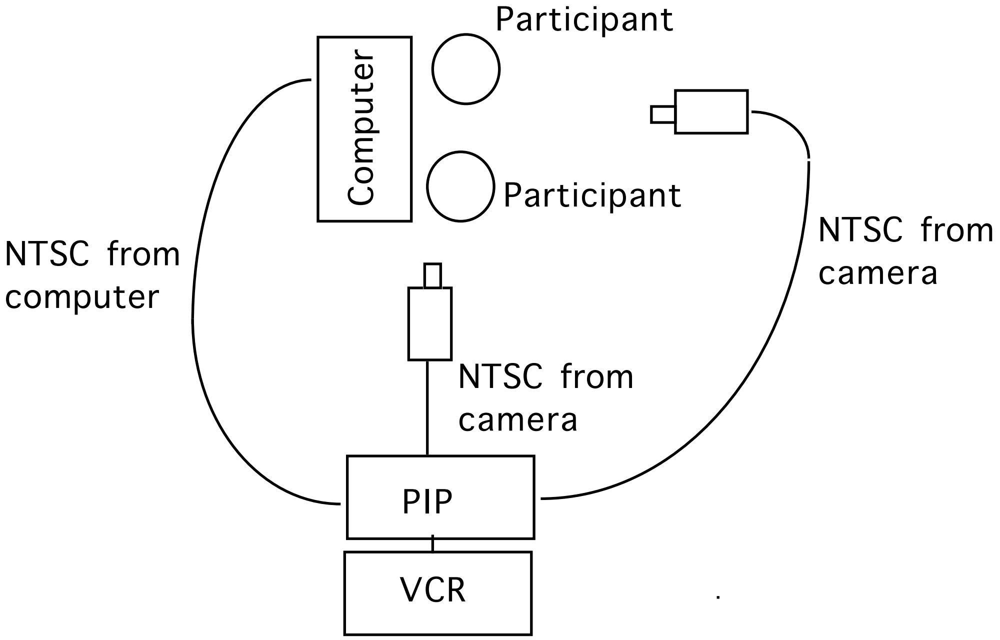

Figure 1: Laboratory Setup

After some experimentation we found that the first camera output from the monitor was quite adequate for analysis and thus dropped the third direct NTSC monitor output since it did not show where people were pointing at the screen. We created a final videotape with a picture-in-picture processor with integrated images from the first and second cameras. The image of the computer monitor with participants pointing occupied the major portion of the final picture, and a reduced version of the long distance image of the two participants was placed in the upper right hand corner. This tape was then viewed by the whole team freely discussing possible problems, interpretations and alternative designs. For some events which we could not immediately interpret, we produced a written transcript which we later analyzed in more detail.

The following is an example from the CVCK development of the richness available from this type of analysis. In this particular episode, after the participants have read a brief tutorial description of how the software works, they are asked to construct a model system that replicates Figure 2. The reader can see that the CVCK interface is a standard tool palette with a central workspace for model construction. A user can select a model component (on the left edge of the palette and from top to bottom) of ventricle, valved pipe, T, straight pipe, elbow, and muscle. A component copy of the icon is dragged into the workspace to the appropriate position. Because components are used in four different rotational positions, a design decision was made to have only one orientation of each component and use a "compass" icon (the cross-shaped double-headed arrows) to create the correct orientation.

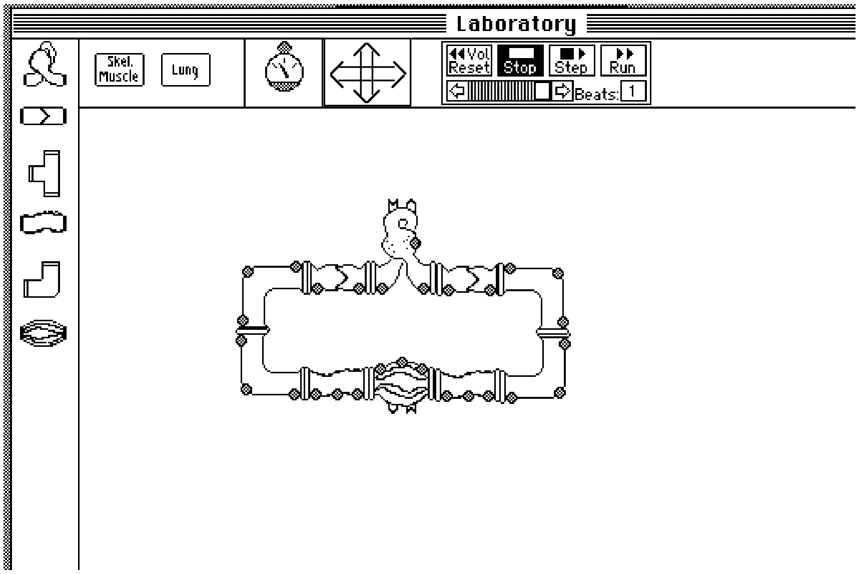

Figure 2: CVCK construction for study

The transcript begins after the participants have already placed one "elbow," i.e. right-angled vessel, component in the workspace. For readability, the transcript has been somewhat edited. The format of the transcript is modeled on Suchman (1987) and the techniques of Wooffitt (1990). (For further details, the article by Jordan and Henderson (1995) on interaction analysis is helpful.) The two participants are "L" and "V".

USERS		THE MACHINE	
I	II	III	IV
Actions not available to the machine	Actions available to the machine	Effects available to a user	Design rationale
L: OK. OK, what do we need (L & V read instructions)			
V: We need another (points to elbow icon on palette)			
	L clicks elbow	palette elbow highlights	add component
one . . . we need two, three more	L drags elbow from palette	elbow added to workspace	place in model (selected object = elbow)

The second elbow is successfully added to the workspace and placed into position. They immediately discover that it is the incorrect orientation and attempt to correct that.

L: No, wait			
How do you //turn this thing around	L moves mouse in circles	elbow moves in circles	change position
V: //turn it			
L: Oh oh shoot			
V: That's right, just move it			
L: No it won't turn			
(twists mouse to turn it)		no effects	
V: OK.			

Here, the participants have attempted a direct mapping of their physical actions using the mouse to rotate the object. The first is to move the mouse around in a circle and the second is to actually rotate or twist the mouse by keeping it in one place. Neither works since in the first place the hardware recognizes moving the mouse in circles as a circular motion of an object on the screen and not a rotation of that object and, in the second place, the hardware does not recognize twisting the mouse as a meaningful action at all. Thus, these two repair strategies fail. (And cannot be used by developers as repairs to the design since they are outside the possibilities of design.) They try another approach:

L: Oh wait			
V: Put it over here			
L: Oh wait, wait. I have an idea. Maybe this works,	L clicks on slider in control panel	elbow un-highlights	deselect component (no object selected)
		slider highlights & moves	change simulation rate
this, wait, where is that thing? here we go	L moves to rotate icon clicks on right arrow	highlights rotate icon no effects	rotate component (no object selected)

This second strategy is to try to find an action that maps to a control icon that will rotate the elbow. Readers should note how the participants use the interface itself as a resource in their interaction. First the slider in the control panel is used and then L recognizes the "rotate" icon which is intended by the developers to resemble a compass. The developers intended that a component be selected, then rotated to the desired orientation by selecting one the four different points of the compass which represents that direction. Note that this is a compound procedure, first select the component, then select the compass point. Unfortunately L deselects the component accidentally by choosing the slider. She then recognizes this and attempts to redo the procedure:

V: OK now, put this one			
	L clicks on elbow	highlights elbow	elbow selected
	L drags it to left	elbow moves	change location
	L points to rotate icon		
	L clicks just outside right arrow	un-highlights elbow (no other effects)	deselect component (no object selected)
L: Darn it!			

L's failure here is interpreted by the developers as a problem of designing the compass points too small. They are very tiny and this user is unable to "hit the target." A possible design change is to make them larger, but then they will exceed the maximum size of an icon in the interface software. L, however, attempts to follow the advice of her partner and abandon the task:

V: Turn it the opposite way.			
L: It won't turn.			
V: Then bring it back over here. (points to elbow icon in palette)	L clicks on elbow	highlights elbow	elbow selected
	L drags it next to elbow icon in palette	elbow moves	change position
	L clicks on elbow icon in palette	un-highlights elbow component	deselect component
		highlights elbow icon	add component
V: Where where'd that thing go?			

This is an unsuccessful attempt to put the elbow component from the workspace back into the palette and start a completely new problem. It can't be done and L tries one last time to get the rotate icon to work:

L:	There.	L clicks on elbow component	highlights elbow	elbow selected
	Oh!	L points to rotate icon, clicks right arrow	elbow rotates to right	rotate component
	There we go. What'd I do?			

Although L does finally complete the rotation problem, the developers have enough evidence at this point to attempt a new approach to rotation of components.

Once we, as developers, had determined the cause of a particular breakdown, it often suggested design solutions. However, and more crucial to this discussion, the repair efforts by the participants often suggested possible design alternatives. Participants' repairs can be found by examining their detailed talk, and by their other actions in which they use the interface itself as a resource for furthering a repair. The first attempts by L & V to rotate the component, namely by direct physical actions with the mouse, were not possible design alternatives. However, we were impressed by L's three attempts to use an icon to rotate the component and thus remained committed to an icon command on the palette. (And did not choose another approach such as a menu based command.) In other words, the design rationale for accomplishing this action closely matched the participants' expectations and intentions. What failed was the actual details of the objects and actions.

We observed three problems with the existing design: a failure to recognize the rotate icon, a failure to select a component before selecting the rotate icon, and a failure get the cursor inside of a very small target. What we finally decided to do was change the rotate icon to a very simple type of icon button labeled with the word "rotate" and which when pressed, rotates a selected component. This new icon is shown in Figure 3. We also decided to add logic for trapping an attempt to use the rotate icon without a selected component. This trap then informs users that a component must be first selected before it is rotated.

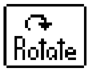

Figure 3: New Rotate Icon

We have found that usability testing, as I have described our usage above, is an invaluable tool in doing design. Though the method takes time for analysis with the whole design team, it also provides an arena and indeed creative environment for them to mutually work out the design in the presence of real evidence about human behavior.

CONCLUSIONS

In a recent survey Nielsen (1993) estimates that 4-6% of corporate and industrial software development budgets are spent on usability engineering. But even the most thorough analysis of usability testing defines it in rather vague terms. Nielsen characterizes it as simply any testing that uses “real” users, not specifically addressing the method (Nielsen, 1993, p.165). And in the most detailed study to date comparing usability testing with other interface evaluation methods, the authors define it as,

> ...*usability testing*, in which the interface is studied under real-world or controlled conditions, with evaluators gathering data on problems that arise during its use. These tests can offer excellent opportunities for observing how well the situated interface supports users’ work environment. . . . The usability tests were conducted by a human factors professional, for whom product usability testing is a regular part of his job.
> (Jeffries et al., 1991, pp. 119 & 120)

These descriptions suggest to us that usability testing is a black art except for the fact that it brings real users into a laboratory setting. This distinguishes usability testing from methods such as beta testing which occurs at the work site where the software is used and does not involve specific tasks or video recording. Using the method described in this paper will provide a systematic approach that has been sorely lacking.

REFERENCES

Apple Computer, Inc. (1987). *Human interface guidelines: The Apple Desktop Interface*. Reading, MA: Addison-Wesley.

Ericsson, K. A. & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press.

Jeffries, R. & Desurvire, H. (1992). Usability testing vs. heuristic evaluation: Was there a contest? *ACM SIGCHI Bulletin*. 24, 39-41.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. M. (1991). User interface evaluation in the real world: A comparison of four techniques. *Human Factors in Computing Systems, CHI '91*. New York: ACM Press.

Jordan, B. and Henderson, A. (1995). Interaction Analysis: Foundations and Practice. *Journal of the Learning Sciences*, *4,* 39-103.

Lewis, C. (1982). *Using the "Thinking-aloud" method in cognitive interface design* (Technical Report RC 9265). Yorktown Heights, NY: IBM Thomas J. Watson Research Center.

Luff, P., Nigel, G. & Frohlich, D. (Eds.). (1990). *Computers and conversation*. London: Academic Press.

Miyake, N. (1986). Constructive interaction and the iterative process of understanding. *Cognitive Science*, 10, 151-177.

Myers, B. (1995). User interface software tools. *ACM Transactions on Computer-Human Interaction*, 2, 1, 64-103.

Nielsen, J. (1993). *Usability engineering*. Boston, MA: Academic Press.

Nisbett, R. E. &. Wilson, T.D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, 84, 231-259.

Suchman, L. (1987). *Plans and situated actions: The problem of human machine communication*. Cambridge: Cambridge University Press.

APPENDIX: SAMPLE MATERIALS

The materials included in the appendix are example documents for a usability test of a new type of pointing device, a foot mouse.

Documents included:

Usability Testing Design Document

Letter to Potential Participant

Screening questionnaire

Informed consent form

Word Processing Task sheet given to participant during testing

Checklist for tester during testing

Observation sheet for tester during testing

Post-test questionnaire

Usability Testing Design Document
Foot Mouse Usability Study

Usability Questions:

1. Can an average hand mouse user learn to use the foot mouse in a reasonable amount of time (1-2 hours)?
2. What physical or cognitive problems do users have learning to use the foot mouse?
   - Guidance to target
     - Pointing in two dimensions
     - Dragging in two dimensions
   - Target selection (button)
     - Single clicking
     - Double clicking
   - Context switching between pointing and typing
   - Repositioning
3. What physical problems do practiced users have operating the foot mouse?
   - Fatigue
   - Comfort
   - Limb coordination
4. Practiced Skill
   - Speed and Accuracy
   - Individual Differences
5. Acceptability by Users as compared to other pointing devices used
6. Reliability of device during tests

Participants

There will be five participants. They will be chosen to represent a cross-section of users—age, gender, experience. Criteria for exclusion include lack of experience with a mouse, Microsoft WORD and Microsoft EXCEL; or problems or disability of any kind with the foot or leg.

Procedures

Selection of Participants.

1. Advertisement in newspaper or from temporary help agency or from student pool
2. Given Informed Consent form and Screening Questionnaire (letter sent)
3. Return Informed Consent agreement and Screening Questionnaire
4. Review Informed Consent agreement and Screening Questionnaire
5. Schedule subject
6. Contact subject with scheduled time

Testing Procedures.

1. Screening test with hand mouse (5 minutes) [Verify subject has skills];
2. Learn how to use the foot mouse (10-15 minutes) [see Learning Protocol];
   - BREAK (2 minutes)
3. Practice tasks (30 minutes) [see Practice Protocol];
   - BREAK (2 minutes)

4. Simple word processing tasks (15 minutes)
   [see Word Processing Protocol];
   BREAK (2 minutes)
5. Answer a Post Questionnaire about using the foot mouse (10 minutes).
   BREAK (2 minutes)
6. Interview and debriefing (5 minutes)

Usability Testing Design Document - continued

Foot Mouse Usability Study

Time for session

Physical time	1 hour (5 minutes with hand, rest with foot)
Writing time	10 minutes
Interview time	5 minutes
Breaks	8 minutes
TOTAL	~1.5 hours

Hand Mouse screening Protocol (5 minutes)

1. Explain purpose of protocol: Verify familiarity with software and hand mouse
2. Three tasks with WORD
3. Three tasks with EXCEL

Learning Protocol (15 minutes)

1. Explain purpose of protocol: Learn to use foot mouse
2. Demonstrate use of foot mouse
    - Positioning of foot on Foot mouse
    - Movement of cursor to a target
    - Selection of target
        - Single and Double clicking
3. Participant practices
4. Demonstrate use of foot mouse
    - Repositioning
    - Dragging
5. Participant practices

Practice Protocol (30 minutes)

1. Explain purpose of protocol: Practice using foot mouse
2. Demonstrate pointing task including error processing.
3. Pointing tasks in two dimensions (127 trials)
4. Demonstrate pointing switching to typing task
5. Pointing switching to typing in two dimensions (127 trials)
6. Demonstrate dragging task
7. Dragging tasks in two dimensions (64 trials)

Word Processing Protocol (15 minutes)

1. Explain purpose of protocol: Using in context of work
2. Give Roberts word processing task.
    - Menu selection tasks (point, drag, single click)
    - Character insertion (point, single click, type)
    - Character replacement (point, drag, type)
    - Word replacement (point, double click, type OR point, drag, type)
    - Delete text segment (point, drag, menu selection OR point, drag, type)
    - Change font point size (point, drag, menu selection)
    - Reposition window contents (point, drag)

Letter to Potential Participant
Foot Mouse Usability Study

PNW Usability Consulting
Suite 418
Eugene Professional Bldg.
132 E. Broadway
Eugene, OR 97403

July 7, 1995

Dear

You are invited to participate in a research study which will determine the effectiveness and ease of learning a foot controlled computer pointing device (foot mouse). I am a human factors consultant who is working with the designers of this device. You were selected as a participant in this study because you answered our advertisement.

I have enclosed two forms for you to read. One is the Informed Consent form. If you agree to participate, please sign and return this form. The second form is a Screening Questionnaire. Please answer the questions to the best of your ability and return it also. A pre-addressed envelope in enclosed.

If you choose to participate, please return these forms by July 15.

If you are chosen for this study, I will contact you by July 16.

You will be paid $10 per hour for approximately two hours of work. My office is located at Suite 418 in the Eugene Professional Building at 132 E. Broadway, across from the Zenon restaurant. Since parking is sometimes a problem in the downtown area, you can park in the large public parking structure on Oak St. and I will pay for parking.

If you have any further questions, please don’t hesitate to call me at 686-3522.

Sincerely,

Sarah Douglas

Screening Questionnaire
Foot Mouse Usability Study

Name:____

Work Phone:________________________ Hours at this phone: _______________________

Home Phone:________________________ Hours at this phone:_______________________

Address:____

Computer System Familiarity

1. Which computer systems do you use?

	Never Used	Use once a week	Use Daily
IBM PC or clone			
Macintosh			
UNIX			

2. How many years have you used a computer?

3. What are your principle uses for a computer?

4. How familiar are you with the following: (Please check appropriate box.)

	Never Used	Use once a week	Use Daily
Computer Mouse			
Trackball			
Joystick			
Trackpad			
Microsoft Windows			
Microsoft WORD			
Microsoft EXCEL			

Physical Performance

5. Are you right or left handed?

6. Have you ever had Repetitive Strain Injury (RSI) or carpel tunnel syndrome with any limb? Please describe.

7. Are you right or left footed?

8. Do you have any problems using your foot? This includes soreness or disability. If so, please describe.

Screening Questionnaire - continued
Foot Mouse Usability Study

General Information

9. What is your age?

10. What is your gender?

11. What is your shoe size?

12. What is your height?

13. What is your occupation?

14. When are you available for two consecutive hours? (Please check appropriate box.)

Week of July 24

	8-noon	noon-4pm	7-10pm
Mon			
Tues			
Wed			
Thurs			
Fri			
Sat			
Sun			

Informed Consent
Foot Mouse Usability Study

For purposes of this pilot study, I understand that I will be asked to do the following:

(1) Learn how to use the foot mouse (10-15 minutes);
(2) Practice pointing with it until I reach a reasonable level of skill (30 minutes);
(3) Use the foot mouse in some simple word processing tasks (15 minutes);
(4) Answer a questionnaire about using the foot mouse (10 minutes).

There will be a break between each task. The total task time will be approximately 1 hour and 30 minutes.

I understand that I will be videotaped throughout the study. One video camera will be focused on my feet, while the other will be focused on the screen. This will help ensure my anonymity. (We need to do this in order to be able to see how well you can learn and use this device. Please remember that it is not you who are under test, but the pointing device.) I understand that I will be interviewed after the study to find out what I felt about the device.

I understand that any information that is obtained in connection with this study that can be identified with me will remain confidential and will be disclosed only with my permission. The Screening Questionnaire that I have filled out will be destroyed at the end of the study. The videotapes, computer data and Post Questionnaire will be kept, but will be identified only by a code number. There will be no means of linking my name to the number.

During the course of the study, I understand that if I find the task too fatiguing or if I feel eyestrain, I can pause for a break. My participation is entirely voluntary. I understand that I can discontinue participation at any time without penalty or loss of benefits to which I am otherwise entitled. However, I understand that I may be discontinued without compensation if I have not accurately represented my skills and information about myself on the Screening Questionnaire.

There are some benefits for me. I will help evaluate an innovative pointing device and I will be paid $10 per hour for my time, paid when I complete the study.

If I have any questions about the research at any time, I understand that I can call the director of the research study, Sarah Douglas, at 346-3974.

I agree to accept all risk of any personal injury that may arise in connection with these activities and release Pacific Northwest Usability Consulting, its subsidiaries, affiliates and employees from any liability in connection with this study.

I agree that I will not disclose any information about the study (including the nature of the study, the nature of my participation, its results, or information about any products involved) in any form, oral or written, to anyone outside of Pacific Northwest Usability Consulting.

My signature below indicates that I have read and understood the information provided above, that I willingly agree to participate, that I consent to the use of data gathered during the course of this study and through the screening questionnaire, that I may withdraw my consent at any time and discontinue participation at any time without penalty or loss of benefits to which I are otherwise entitled, and that I will receive a copy of this form.

Signature____ Date____________________

Word Processing Task
Foot Mouse Usability Study

1. Create a new document

2. Type the following paragraph:

He sighed. And that was the end of the discussion. Over the years that we've been married, we've learned to sidestep the subject of my family, my duty. It was once the biggest source of our arguments. When we were first married, Phil used to say that I was driven by blind devotion to fear and guilt.

3. Insert a comma after "Over the years"

4. Replace the comma after "my family" with "and" .   Use dragging to select the comma.

5. Change the word "biggest" to "only" .   Use dragging or double clicking to select.

6. Delete " that we've been married," .   Use dragging to select the text.

7. Insert a title at the beginning of the paragraph "Excerpt from Amy Tan's The Kitchen God's Wife". Center this title and make it bold with a larger font point size.

8. Scroll the window down.

9. Save the document with the name "temp".

Checklist for Tester
Foot Mouse Usability Study

Explain Purpose of Study
- Develop a more usable device
- Testing device, not you
- Can stop experiment at any time

Adjust chair height and comfort

Learning Protocol

Acceleration Curve_______________

Pause on?: ________

Foot Positioning and Comfort
- Dominant foot
- Wedge

Moving
- Keep foot flat
- Relax leg
- Don’t raise toe
- Don’t raise heel
- Move whole leg for gross movement
- Move heel to pivot for fine movement

Clicking
- Raise and lower heel

Double Clicking
- Raise and lower heel twice

Dragging
- Move with heel up

Fine Positioning
- Use shift key

Moving device on floor to reposition
- Use F6 key to freeze cursor

Observations

Foot Mouse Usability Study

Participant: ________________________

Physical Problems

- Guidance to target
    - Pointing in two dimensions
        - Gross Movement
            - Moving with heel up, not flat?
            - Moving with foot, not leg?
        - Fine Movement
            - Pivot with heel?
    - Dragging in two dimensions
        - Gross Movement
            - Moving with heel down?
            - Moving with foot, not leg?
        - Fine Movement
            - Pivot with heel?
- Target selection (button)
    - Single clicking
        - Using toe?
        - Clicking by raising heel, not down?
    - Double clicking
- Context switching between pointing and typing
- Repositioning
- Fatigue
- Comfort
    - Shoe Type and Size OK?
    - Used wedge?
- Limb coordination
    - Used which foot?
- Other

Performance

- Speed and Accuracy
    - Small targets OK?
    - Long distance OK?
- Individual Differences

ACCEPTABILITY

- Compared to other devices

RELIABILITY OF DEVICE

OTHER COMMENTS

Post-test Questionnaire
Foot Mouse Usability Study

Learning

1. Did you find this device easy to learn? Explain.

2. How would you compare learning the foot mouse to learning the regular hand mouse?

3. What did you find hardest to learn about the foot mouse?

Practiced Skill

4. At the end of the study did you feel like you could control the foot mouse in all situations? If not, where did you have problems?

5. Did you find the foot mouse comfortable to use? If not, why not?

6. Do you have any fatigue from using the foot mouse? Where?

Overall

7. If you had the opportunity to use a foot mouse with your computer, would you? Explain.

8. Would you prefer to use it for certain tasks? Examples: word processing, spreadsheets, video games.

9. Would you like to be able to switch between the hand mouse and the foot mouse?

10. Can you imagine any circumstances in which you or someone else might prefer the footmouse over other pointing devices?