

Origins of the Apple Human Interface

Larry Tesler
Chris Espinosa

5:30 PM, Tuesday, Oct. 28, 1997
Computer Museum History Center
Building 126
Moffett Field
Mountain View, CA 94035

This is a verbatim transcript of a public lecture given on October 28, 1997.

This transcript is Copyright © 1997 by Computer History Museum and is for private individual use only. It may not be published, in any form, in whole or in part, without prior written permission of Computer History Museum.

LEN SHUSTEK:

[Initial portion not on videotape] ... eventually opening a public museum, focused on the history of computers, hardware, software, documentation and so forth, designed for adults, not a kids' hands-on museum, but it's an adults-oriented museum. It's a long term project; we're probably going to take four or five years to get there: to build a building, to find a location, and site, and so forth. In the meantime we're focusing on continuing to build the collection, which is the finest in the world. It's got thousands of artifacts, and thousands of photographs, and documents, and books, and so forth.

We're inviting people to help us with this project by doing lots of things. We have weekend volunteer days, when people come and help us work on the collection. We'd like as many people as can to become members of the museum, and you can do that in several ways. One program, that we just started, is that you can become a member for a hundred dollars, and get a card that gives you access to the History Center, and also gets you access to Moffett Field. So on Wednesdays and Saturdays you can, without going through the process you went through today, get onto the Field and to have access to our visible storage location.

In addition, for about the next six months, you have an opportunity, one-time only, to become a founding member of the History Center, by donating a thousand dollars or more, and be permanently enshrined as one of the founders of the Computer History Center. It's the same kind of program that was done by The Computer Museum 10 or 12 years ago when it was in its founding stages.

What else are we doing? We are mounting a bunch of exhibitions in various places. We've done several at trade shows and conferences. We've got one that's going to be opening next week at the new Gates Computer Science

Building at Stanford, where we're taking part of our collection and showing it off. And until we have our own permanent space, that's a good way to show off some of our wares.

We've got more stuff coming. As we speak, there are four tractors being loaded in Boston and heading across the country with the next installment of the machines and part of the artifacts from Boston. Included in there is the JOHNNIAC, and the CM-5 Thinking Machine that we just got from MIT, and a whole bunch of other very valuable artifacts.

If you want to get more information about the Computer History Center, talk to me or Gwen Bell or Joey or Dag, or any of the other people who are on staff. We'd like to thank the people at NASA Ames here for their hospitality, both in the use of this conference center, but also in the use of the warehouse space, for which we're very much in debt to them. There are some flyers in the back that you can pick up, that describe what's going on here, both NASA generally and their Center for Excellence in Information Technology.

This is one of a series of talks that we've been giving, in cooperation with the Bay Area Computer History Perspectives. Peter Nurske runs that, and he will introduce tonight's speakers.

PETER NURKSE:

Thank you, Len. I don't really run the Bay Area Computer History Perspectives. It's more a joint venture between myself -- I work at Sun Microsystems -- and Jeanie Treichel, who also works at Sun Microsystems. Jeanie was at Xerox PARC, many years ago, as part of history herself.

We started this computer talk series over four years ago, back in [1993], and two of our speakers at the first program are here in the front row: Norm Hardy and George Michael, from Livermore. So a lot of people have been coming back over the years, from one program to another. Larry Tesler, who's one of our two speakers tonight, himself did a program along with Bob Herriot, Larry Breed, Don Wise, over three years ago, on the Stanford card stunts, which I wrote up in the description [as] the largest bit map display ever. You know, it was like several thousand students holding up colored cards in sequence. That was a feat itself, to do the programming of instructions, back then at that time.

Larry has also been coming to many of the programs before and since then, so it's like this history talks series had developed a history of its own. So, tonight we have Larry and Chris Espinosa from Apple, talking about the early Apple user interfaces. I think, as you may have heard, Larry came to Apple from Xerox PARC, back in 1980, and he figured he was bringing a lot of user interface knowledge with him, only he's found out recently that user interface testing and development existed at Apple before 1980. And Chris had a part in that. Chris -- I

think this is correct -- was probably the youngest Apple employee ever; [he] did demos at Apple while a high school student.

So, Larry

LARRY TESLER:

We're glad to be here. This is a talk actually that's based on one that Chris and I gave at Apple back in July. We got clearance, thankfully, from the Apple lawyers, which came about two - three weeks ago, so we could give it here, just in time to announce it. We're grateful to Apple to release this for public disclosure, because we think it's of general interest.

Asking about what was early user interface design at Apple was kind of like asking about early engine design at Mercedes Benz or something. Everybody was doing it. So, you could have gotten a lot of people up here to give a talk, and you get different perspectives on the same things that happened. So today you're going to get the perspective of a couple of us.

To get it to be a little more grounded in something more [closely related] to fact than opinion or perspective, we're trying to use as much as possible actual, original documents that were created at the time. So you see actual, contemporaneous thinking as it was going on, instead of just our memories, which fade over time and tend to aggrandize over time, and so on.

Here's something which is not from Apple. This is the cover of Byte [Magazine], from August '81, that was the Smalltalk issue. Colorful -- this one doesn't have color, but it was a very colorful balloon. The reason I show it is that there was a lot of influence from Xerox obviously on the user interfaces that were done on the Lisa and the Macintosh, but maybe not as much as some people think. The media tends to say, "Oh, they just copied it." So I'm going to show you what it looked like, to be working on, from that article, which I wrote, to be working on a Smalltalk, at that time, 1981. You see what the windows looked like, kind of a little rectangular title, scroll bar without any arrows, and a pop-up menu. There were two pop-up menus that you could use for each window. So, it was a pretty different looking interface.

One of the principles that was in the Smalltalk environment was to have no modes, and this was actually of a tee shirt that was made for me by a friend, that said "Don't Mode Me In", that represented sort of this mission that I was on, to try to eliminate all modes from user interfaces. Which isn't necessarily a good idea, but it was definitely the mission I was on. [Laughter]

And at the end of that article, it mentioned that Apple Computer had gotten a license to Smalltalk, and was maybe going to do something with it, but maybe not going to do anything with it. In fact, Apple, and Hewlett Packard, Digital

Equipment, and Techtronics all did do various amounts of Smalltalk development after that. But it was really independent companies that managed to make Smalltalk somewhat of a success.

I'm jumping to the end here before I go to the beginning, just to let people know what the Lisa was. The idea of the Lisa was to have something that was very, very easy to use. People loved the Apple II, which was what we had at the time at Apple, but it took a long time to learn how to use it. What we wanted to end up with was something that was easy to learn, and just to show you that we did do that, this is a study done at the end, where there were 18 subjects: 6 using LisaGraph, 6 using LisaWrite, and LisaList. LisaGraph was a business graphics program; LisaList was a list manager, kind of a flat file database; LisaWrite was a word processor. And as you see, for all but one user using LisaList, it was under 20 minutes to get through the core competency to use the application. And if you ever had a chance to use a Lisa, and I hope there will be one operating here someday, then you'll see that it was extremely easy, and in many ways more bullet proof, I'd say, than the Mac. Not simpler than the Mac, actually, but we loaded a lot of things into it to try to make it so people couldn't get themselves in trouble.

Okay, so let's start at the beginning. I started at Apple on July 17th, 1980, and I just left there a few months ago. I was there for 17 years. This memo was written on July 18, 1980, so one thing I discovered when I left the big Xerox Corporation and went to this little Apple startup, was that you just really get engaged immediately and they put you right to work. One thing I was told, the day I got there, was that the user interface design was basically done, and they were just going to write it up and finalize it, and implement it, and that was the end of it. And it was too bad I took so long to show up at the office, because they were hoping that I could have contributed to it, but now there was no time left to do that. And this was the 18th, and they told me that all the decisions had to be made by the 23rd, and by one week later the external reference specifications had to be done, which was the complete spec, but we couldn't compromise quality at all, and so, what could I do between the 18th and the 23rd, to make sure we made all the right decisions?

So I started explaining how it was really necessary to do months of user testing, and careful iteration, and I gave my whole spiel about the right way to design software, and they said, "Well, sorry, we only have five days, and the weekend in there." [Laughter] So, we got together and we came up with a compromise, which was to lay out a schedule that was very aggressive, and people promised that they would meet their part. So let's go through the schedule -- this was the 18th.

So here was the schedule. Monday, we're going to run user tests, so -- well, we're all going to try it ourselves. Everybody's going to try Bill Atkinson's prototype. Bill Atkinson was the person building the prototypes at that time, and thinking up a lot of the ideas. And we were going to evaluate several issues that

were going on. Then we were going to bring in, after two hours of that -- we were going to take a break for lunch. Then at one o'clock we were going to bring in Sue Espinosa, who was in charge of training, and also coincidentally is Chris Espinosa's mother, and then she was going to give her opinions about that, and maybe help us decide on what would be easiest to train in her opinion. And then at two o'clock we were going to discuss, for an hour, a couple more issues, and now we would be done. We will have discussed all the issues. And in between three and five o'clock we would present this all to Steve Jobs, and Steve would make the decision, which we knew would be definitely forthcoming. [Laughter]

And then on Tuesday we were going to take the things that Steve liked, and we were going to test them on subjects, from 9:00 to 10:00. Two subjects. [Laughter] Two more subjects from 10:00 to 12:00 -- we must have figured we'd be slowing down by then. [Laughter] 1:00 to 3:00, we'd evaluate the test results.

So that was it. Remember, that was supposed to be July 22nd. For the next 15 minutes or so, keep in mind that that was the schedule for July 21st and 22nd. And we actually did that. We did all those tests. We did talk to Steve Jobs. He gave his opinions. We did some more testing. And then Bill went to write up a user interface specification, external reference specification. I was a little worried, because we hadn't really done extensive testing, but we tested a couple things.

So this date, August 6, which as you may notice, a few days later than the deadline date that we had provided, but it took a while. And it said that you had one week to review and return your comments. So we were already starting to slow down a little bit here. And let's look at some of the names of the people who got this memo, because it will turn up to be important later in the story, or just interesting right now: Ken Victor, who is going to respond to it later, and Jef Raskin also. Barry Smith. In addition to that, notice we sent it to Mike Markkula, who was one of the founders of the company, and Steve Jobs. Trip Hawkins, who was the head of Product Marketing. We sent it to everybody in Lisa software group. We only sent one copy to Lisa hardware; they get to share. [Laughter] We sent it to every vice president we could think of, and to two people who became Apple Fellows somewhat later, Rod Holt and Steve Wozniak, who's here in the back today. I don't know if he remembers receiving this memo, but he probably does, it was pretty dramatic.

I think we did some things right. Bill started out by talking about who is the customer. So it wasn't just some easy to use computer, it was an easy to use computer for specific people. So, what was it? First of all, it's a single user workstation. That was an important point in those days. Well, it's still an important point if you work at Sun or something. And [it was] designed to enhance the productivity of the office worker. So our market was office workers. And it went on and talked about the hardware and so on.

It's more than a piece of hardware, though. It includes major software, and we had specific applications in mind: word processing, graphics, data management, and communications. It wasn't intended to run thousands of applications, not the Lisa. The concept that Trip Hawkins had was, they had done studies and they had found that most people, on the Apple II, ended up using only two or three applications. Therefore -- this was the great leap -- all we need to do is provide two or three applications, that's all anybody will ever want. It was an interesting theory, and Apple funded a whole project based on that assumption.

And then it said, "The software is integrated through a powerful and simple user interface." And it doesn't say it here, but John Couch, who was in charge of the Lisa project, did a sort of Kennedy we're-going-to-send-a-man-to-the-moon-and-bring-him-back thing; he said "All I want is, I want to be able to go into a spreadsheet, put some numbers in, go and graph them, take the graph, paste it into my word processor document, and include it in a report." He didn't say "paste" -- you know, "put it into my word processor document." He said, "[If] I can do all those things in an integrated way, then we'll be way ahead of anything available on the Apple II, where there are all these separate programs that don't interact with each other." That's all he wanted. Whenever we brought him any issue he said, "Remember what I want," and he went through the whole mantra. And that was a very healthy thing. It kept us focused. And then we also knew when we were done.

Well, who were the users? Non technical, non analytical people. Non analytical. That's a very important point. First time users. Well, in 1980 there were a lot of first time users. Almost everybody. And here were examples: administrative assistants and secretaries. Also, secondarily, managers, accountants, and executives. It wasn't for programmers, it wasn't for kids, it wasn't for people at home doing recipes, this was an office system. And so, when you start looking at it, and comparing it maybe with the Mac, you'll see some differences based on that. And you may see some similarities to Windows.

Design philosophy: so there was a whole section on the design philosophy, and what were the most important features to put in? Graphic images. Well, that was pretty vague. Menus. A pointing device that you could move around freely, as opposed to the cursor keys we were using up to then. And, we were going to let you invoke commands from the keyboard also, because there were times when you would do things over and over, and people felt that sometimes the mouse slowed you down a bit.

So let's look at this keyboard. This was the keyboard that existed on the hardware when I arrived. It had a Code key in the upper left corner, and some of the keys, on their fronts, had some characters. In other words, in order to [get] the keyboard width they wanted, they couldn't fit all the characters on the keys. So they squeezed a few extra characters on the fronts of the keys, and had a

special key to invoke those. And later we decided we could use that key together with other keys to do another function, that you'll see later.

Now, I didn't like the hardware design too much, and I wanted to change it, but they said "You can't change that. You know, you came from a research place. We just have to ship the hardware that we already built, otherwise we won't be able to ship this product when it's going to go to market." I said, "Oh, when is that?" They said, "Six months." That was 1980. To jump ahead, it didn't ship 'til the middle of '83. Probably my fault. Okay. [Laughter]

So, here was the pointing device. Kind of a brick -- that's a mouse. And, the color is me; I added color last night, just to highlight things you should look at. But it had two buttons at that time. And I handwrote on here something about the shape. I wasn't too happy with the shape. And I wasn't sure what distinguished these two keys, and whether there was a little space between them, or whether they were flush, or what. So, you'll see some of my handwritten scribbles on these memos.

And here's how the mouse was used. Well, actually they didn't know a lot about how the mouse was used yet. They mostly were focused on the window manager: how the user would manage the windows on what we now call the desktop. And so at that time the idea was this: there would be a click and a drag, and those were two things you could do. You could click and let go, or you could click, hold, drag the mouse, and let go. Sometimes we called it "drag," sometimes we called it "draw," various names. And then the buttons got names: the "point" button and the "grow" button. And depending on where you pointed, and whether you clicked or dragged, and which button you used.

This page here shows what happens if you went into the title tab of the window, the little thing that had the name. And it either would display the window menu, move the window, open or close the window, or change the size of the window. All of it done by clicking in the title. So there was no "grow" box in the lower right, there were no menus, it was all done with just one of two mouse buttons and a click or drag operation. Now, there was a menu at the bottom of the window, but it had nothing to do with window operations. It had to do with operating on the content inside the window. We'll get to that.

I should stop saying "when I got there" because by then I'd already been there for two weeks, so I had some influence on this. I can't remember what. So here's a letter to Jef. Now "Jef" is misspelled. That's Jef Raskin, who does spell his name that way. And he became the person who we were always sending these letters to, on the Lisa project. And, as you see, this scroll bar was on the left side. It had the horizontal scrolling arrows at the top, the vertical scrolling arrows at the bottom. Now, those of you who know the Next machine know that that's a feature, is that you have the two arrows close to each other so you don't have to move very far to go between the two arrows. That was actually designed in 1980

by Bill. And then at the bottom, the edit menu had Cut, Paste, Dupe to copy things, and Undupe. And then on the side there were the equivalent of today's icons. This was taken from Smalltalk, where you took the tab of the window -- threw away the window, and just kept the tab -- called it collapsing the window. And these were windows that were available but closed right now. And you could open them up.

Then we had this idea of fill-in menus. Just to jump ahead, they were what we call dialog boxes today. And there were two kinds. There were kinds where you somehow open a sub-menu and you get to type into that sub-menu -- don't ask me exactly how. The other is where it actually opens up a big box, like we know today. That was this thing called forms. And so it was kind of a vague notion at that time, no one had actually drawn any pictures even of what those looked like, I think. And again, we said that menu commands can be invoked from the keyboard, and I put a little note saying "Disagree somewhat." And I wish I could remember what I disagreed with. I think I was afraid they would do too much keyboard stuff.

There were scroll bars. You would hold a button down to scroll continuously. Now, there was an interesting idea here that you would double-click to flip to the next page. Anybody that knows about user interface design with single and double clicks -- usually you make it so that when you click once, you can start doing something, and then when the second click comes, it doesn't sort of negate what you already started to do. This is kind of interesting, because you clicked once, it would start to scroll a little, and when you clicked a second time, it would realize you wanted to jump a lot. So you'd get a little motion, and then a jump. That actually might have been kind of cute. I don't think we ever implemented it.

Then there was something called the elevator shaft, with the elevator in it. And that was what is still today, that little box in the scroll bar that you see. There was the abbreviations window, that was that Code key. We had to find a use for it. So you could type Code, and then hit a letter on the keyboard, it would look up in a table and it would expand your abbreviation. This was the kind of thing you'd find in a Wang word processor. So we put that in, because this was for secretaries.

Then there was something called the system error window. And that's what you now know as alert boxes, in Mac and Windows. And these were things that would come up when there was a problem, and it would make some kind of beep, and would tell you what the problem was. So I put a little question mark here. Is the system frozen when that comes up? Can you do other things? And we eventually decided it was modal, but actually that took quite a while to decide.

Now, we talked about testing. We said that the design's already based on studies of user reactions to various models; as you saw, we did three or four hours of testing. [Laughter] And we're going to do more testing, and avoid any glaring

design flaws. I was happy that that was put in, because that meant we could go and fix some more problems that I was sure were there, because we hadn't done enough testing yet.

So, what did it come down to? This is what it looked like: the basic model was overlapping windows, which is right from Smalltalk. There was that box at the top, just like in Smalltalk, that little title bar. The scroll bar was on the left, just like in Smalltalk, only instead of just that little box, there were the scroll arrows there. We thought it would be nice to have something to point to. In Smalltalk, you push the button and then the arrows appeared, and we thought that was a little weird. And there was this menu at the bottom. Now, if any of you -- who's ever used the system called UCSD Pascal? If you remember, there were menus in UCSD Pascal, and you would type the first letter of the menu name, and it would bring up a sub-menu, and you'd type the first letter of menu name, and you go through a hierarchy of menus.

Now, also there was a machine -- I'm going to forget the number -- a Hewlett Packard computer, 2000? [Inaudible comment] No, it was a later model than that. HP something thousand, that had a row of function keys on the top of the keyboard, and at the bottom of the screen, under each key [comment: still have it?] Still have it. And you'd hit that key, that would activate that menu item. These were the two products that the authors of this interface knew the best. They used UCSD Pascal to do their software development, half of them came from Hewlett Packard, and so this was kind of a blend. And so they had function keys at the top of the keyboard that you could hit that would invoke the corresponding things, and they also had these hierarchical menus that were straight out of UCSD Pascal. Not the little pop-ups that were in Smalltalk. So that's where it was in August.

Cut and Paste were two of the commands in the menu; that came from Smalltalk. And, as you see, there was a wastebasket. That's what you call the clipboard today. We used to call it the wastebasket. Different from the trashcan, which was a confusion we had later, and that's why we changed the name. But when you cut something, it went in the wastebasket, like in the office, you cut something and throw it in the wastebasket. But then you could take it and paste it somewhere else, and it was sort of a weak metaphor. And so you'd select it with a mouse. Notice that when you selected, there's this little arrow here. That turns out to be a significant point later.

I'm moving fast because there are a lot of slide here. Now, Ken Victor, remember -- is Ken here today? No? Ken was one of the reviewers of the document, and he provided some input, very quickly. And he said, "Why can you only select with a mouse?" He must be someone who used EMACS or something; he wanted you to be able to select from the keyboard. You know, fourth line, third word, seventh character -- should be able to specify what you want. And fourth occurrence of the word Lisa, you should be able to say things like that. And he was maybe

either -- he sort of didn't get it, or he was years ahead of his time, whichever way you want to look at it. But this is what he would like to use, and we had to sort of explain to him that our target market didn't want to find the fourth occurrence of Lisa in the third paragraph. They could see it with their own eyes, and select it with the mouse, and that was just fine.

Now, here was something I wish we had listened to. At this point, it said that you were growing windows from the lower right. I showed you something before showing that it grew at the top. I think in this document it was sort of ambiguous, whether you grew it from the top or the lower right. And he didn't like that idea. He wanted to be able to grow from any side. I do think that's better, and we didn't pay any attention to him. They were worried about the screen space being too tight, or something.

Moving a document within a window: he wanted to know, whether we could do what we now call live scrolling. As you move that elevator up and down the shaft, he wanted the text to move continuously. Problem was that the Lisa processor ran at, what, a quarter? [Comment: Five megahertz.] Five megahertz? It didn't even seem that fast, to me. Couldn't do it. So, some things were just pragmatic.

He didn't like the idea of Cut and Paste. He wanted us to use Move, Copy and Delete, and Transpose. In other words, say you have two things you would select, move this to there, copy this to there, transpose these two things. Well, this wasn't the first time I'd heard that argument, and in fact this is one of those religious arguments like one button or -- one versus two buttons. Do you want Cut, Copy, Paste? Do you want Move, Copy, Delete? The reason we went for -- well, you can argue for either one. It is two steps to do Cut and Paste. But there are a lot of advantages to it. One is that Move and Delete sort of start with the same thing. Some people think that's good, some bad. Another reason is that you don't have to be able to see the destination when you are copying or cutting the source. That's the most important thing. And on a screen of limited size, when you have windows overlapping, it's sometimes very hard to get them all lined up, so you can specify two targets. Where you have windows popping up and down, you get very confused.

The other thing is, that I had a secret agenda, which is, I thought the machine should be used not for what they talked about, office systems. I thought, well, that was good, but I don't want it to be used just for that. I thought it would be a great machine for publishing, and that we'd be able to do Cut and Paste to do page layout, which was my own personal interest. And so I was advocating that because that's definitely the way you'd want to do page makeup.

But we did user testing, and the users slightly preferred the cut and paste model. The wastebasket window was able to not only have one thing you cut out, but multiple things you cut out, and he had some comments about how to deal with

the multiple items. Some of us thought it should only hold one; he thought it was okay to have multiple ones, and he lost in the end.

Okay, here are some comments from me. I was complaining because nowhere in the spec did it say what the response time should be. It said all about the user, what they'd do with it, but it never said that this needs to respond within half a second, this needs to respond within two seconds, that needs to respond within ten seconds. It never said it anywhere, and I was worried if it didn't say it, the engineers wouldn't do it. And if any of you have ever used a Lisa, well, it was really slow. I lost that argument. They said "It'll be as fast as we can make it." And that was too slow.

And then it talked about a really interesting question, which was, well, we want hierarchical menus, and we want keyboard accessible commands, so we seem to be fixated on the way they do it in UCSD Pascal: you type a key, get another menu, type a key, get another menu. And that just bothered me, and I thought there's a lot of mistakes going to be made. People are going to forget which level of the hierarchy they're in, how they get back out of it again, it's just going to be too complicated. And I didn't like it.

So -- we'll come to how that was solved later. The other thing I was complaining about was the two-button mouse. We didn't use the second button for much else for a lot of different things. We didn't use it for much. And when we did, there was no way to remember which button did what. And so, I wanted to eject the second button and just have one. And that's August 13th.

Operations on windows. I proposed here that we have this box in the lower right hand corner, that we used for growing the window, instead of doing it from the top. And also I thought there ought to be a way to split a window into panes. And then I had a whole section here about moving a document within a window. I go and talk about the scrolling and why I don't like the way it is. And then I said, "Why don't we put one arrow at the top of the scroll bar, one at the bottom of the scroll bar?" And I actually prototyped it.

Those of you who know the Xerox Star know that the Xerox Star had that feature. Those of us who were from Xerox were under a confidentiality agreement, that we weren't allowed to tell the other Apple people about the Star. And so, as you go through this, you'll sometimes find that things seem like, "Isn't it obvious solution? You could have got it from the Star." But none of us were able to say how the Star was. We were only allowed to comment on other people's proposals and make new ones. And it was an interesting kind of dynamic. But, gradually over time, a few Star-like things crept in, but not many until the Star actually shipped, or it was announced, in June of '81. But here we are in 1980.

Then I suggested we try an I-shaped cursor for text selection, instead of an arrow. It actually is something I had done in my early years at Xerox, in a

particular word processor I had done. I also said I wished the cursor would be bigger than 16 by 16, to go to 32 by 32, which we eventually did.

Menus -- this was what, August 18th? "I think we should experiment with menus at the top or right of the window, or even totally detached at the top of the screen, full width." So up to this time, we were doing sort of the way Windows does, and every window had its own menu. And I was thinking, "You know, maybe we ought to put it at the top." You'll see why in a minute when you see the pictures. And then I said, "Hierarchical menus -- well, maybe we have to have them, we can't get rid of them, but could we make them mouse accessed instead of having to type all these keystrokes?" So, these were my two causes of the moment.

Barry Smith, who was in Product Marketing, responded about the same time. And he said -- at this point, Bill had moved the menu bar, that Edit Cut Copy Paste thing, from the bottom of the window to the top of the window. That was the change. And Barry liked that. He liked it a lot better near the top. Called it "near the top." And then he wanted to know, though, what would happen if there were a lot of menu items, and could the menus scroll. And those of you who know the history of the Mac user interface, know that at some point, around 1990 or so, we started adding scrolling to Mac menus. But we didn't have them in the early Mac or the Lisa, and I was against it because that would have encouraged people to have a lot of menu items, and I didn't think they should. So I kept that from happening. You can't constrain people, though.

He also did not like the fact that you had to type multiple keystrokes to a hierarchical menu, and so he was thinking maybe we could get to something where it was a single keystroke to invoke a command no matter where it was in the hierarchy. And I think Bill was experimenting with that at that time. And he also didn't like the two-button mouse. He thought it was confusing.

So, August 18th -- I guess that was August 13th, before. A few days later, after intense political, polite discussions, heated sometimes, Bill and I decided "Hey, let's get rid of the second button. Let's shake things up. One-button mouse." So we wrote this memo together, so that people knew we were real serious. Sent it to the usual suspects. And we say, "We recommend a change to a one-button mouse." And whoever wrote this comment, and I'm glad I don't know who it was, said, "Well, maybe, not sure that's a good idea." But it turned out that Trip Hawkins, who later went on to found Electronic Arts and 3DO, was the Product Marketing manager. He loved it. He said, "That's what we've got to do. We've got to think more kind of consumer oriented, really simple, get rid of all this computer science concepts, and give people something real simple." He liked the one button.

So after that, we had a period, sort of, of good will, where we were able to do user interface design without a lot of complaining. Which ended after a while, as you'll see.

Well, if you only have one button, some things are hard to do. You can drag a short selection, but what if you want to select something three pages long? It's a long time to drag from one end to the other. So, we thought, okay, well, we'll use the Shift key on the keyboard, because this doesn't happen very often. You click the beginning of the selection, you'll scroll to where you want to go, hold down the Shift key, and click the end of the selection. So that's where we got the Shift-button thing. Adjusting selection.

And then we said, "The one-button mouse is superior. Look at the reasons: superior human factors, you don't have to think of a name for each button." [Laughter] That's important, when you're trying to ship product in six months. The mouse will be cheaper to make, because one less button -- I mean, just think of what the cost of that little switch is. And, something a little more profound, someday when we start having other things than mice on our computers, like joysticks for games, tablets, touch screens, we'll be glad we only had one button because that one finger or whatever or pen that the user is using is the button. That turned out to be sort of prophetic, although very few such pointing devices take advantage of that, in fact. They usually have buttons anyway.

Okay, so August 22nd, this is a revised version of the user interface, and it says, "This is it, now. We're done." So now, instead of a week, you only have three days to return comments, because we must be done. There couldn't be more than one or two comments; they're probably just typos or something, by this point. We'd been working on this interface for a month, already.

Okay, so what did it have in this version? It had a concept called "dimmed highlighting." This was introduced for the first time, that I know of, here. That if some of the menu items, instead of eliminating menu items that weren't relevant, which -- we would dim them, we would gray them out. Instead of being black type they would be kind of in gray type. And then we said, "They can be invoked any way the user wishes, although they wouldn't do anything." And we were also going to use dim highlighting for another purpose, which was to show toggles. And as you know, that was later changed to have checkmarks, instead of just using this for two purposes, because that was very confusing. But at that point we thought we'd get away with one mechanism for both things. And it was Occam's Razor: if one mechanism will do, then don't have two. Well, it turned out that one didn't do, so, we had to apologize to Occam. [Laughter]

Now, as you know, you can do command keys on the Mac or Lisa keyboard, and you can invoke commands from the keyboard. And we knew it was important to reserve some for the most common commands, because it might be okay for an application to use Command Shift J for something, but we wanted to be sure Cut, Copy, Paste, Undo were the same for everybody. Bold, Italic, Underline, and Normal -- we thought these would be used the most. And so, the whole sequence -- I'm not going to go through them all today, but we had various theories, and as you see here, it's rather different from what we ended up with. Z

X C D was sort of the order on the keyboard; Cut Copy Paste Undo was the order on the menu, so that was it. Over time we refined this. This led to many very heated arguments. People would rather argue about trivial things like this than important things, but we did have fun arguing.

Does anybody remember what they really are? Today? [Comment: Control-I, Control-V] That's right, Control-I, Control-V, and then for Cut? [Comment: Control-X, the same as Wordstar!] Wordstar? Oh, I didn't know that. [Comment: That's part of the problem of most computing, is people don't realize what...] Well, we did X because it was a cross-out, and we did V because it pointed down like this, and you were inserting, like an upside down caret. [Laughter] That was Paste. And Z was the closest one, because we figured you'd Undo a lot. And C for Copy, that was easy. So that's how we did that.

So this was the anatomy of a window. We had title tab, and notice, next to the title tab, we now move the menu to the top. As was commented upon, someone obviously had got a sneak preview draft, and that earlier memo. And look, there are menus that are popping down from the menu bar, first time. When it was at the bottom, it didn't make a whole lot of sense, but now that it was at the top, Bill could, instead of having the menu come here and replace the previous menu, like it did in UCSD Pascal, he could use the second dimension and have a menu pop out to the bottom. Don't ask us how we're going to do third-level menus, but at this point, this was good enough. We had two levels of menus, and we were happy.

Over here, there was the elevator shaft. Now we called it the thumb instead of the elevator, because we wanted to have horizontal scrolling too, and elevators in those days didn't go horizontally, although Scientific American says they're about to. And then we had continuous scrolling arrows, and notice the arrows point this way, but when you click on these arrows, the text was going to move that way, as opposed to the other way, which is what it does now. And that was an interesting argument, that I think I have covered here too.

We had active folders, so we had to have passive folders; the folder that wasn't active was passive. Now look at the bottom. Who has ever seen Mac OS8? And when you drag a window to the bottom of the screen it just kind of pops, and becomes just a tab, and then you can click it and it will slide open. Guess what? That was in the original design, in August 1980, that we were going to do on the Lisa. And it only took 17 years to implement. [Laughter] [Comment: Microsoft's doing a similar thing.] Microsoft did something sort of similar, but not really the same, in between. There are some differences, but I won't comment right now.

So what happened if the window was narrow? Well, if the window was narrow, this shows that the menu wouldn't fit; it would stick out. And if it was over on the right, it couldn't stick out on the right. It would stick out to the left. If it was too close to the bottom of the screen, then it would do this. And so, Bill showed all

the cases, and a lot of us weren't too happy with this. So I went back to Bill again, and I said, "Bill, the top of the screen -- top of the screen!" None of us liked the idea of a two-line menu, which is the way Microsoft solved it, in some version of Windows.

Here's an example of a menu item being selected. Notice that he's now got highlighting of the menu title, and he's got highlighting of the menu item. Bill had a long night one night and came back with that. And we had the help menu in here.

The way we operated was, that Bill spent every night programming, and the next day we would do testing and arguing. I don't know when he slept, actually. Did he sleep? And then he would do it again, every night.

Here's a memo from Gail Pilkington in Publications. The people writing manuals for the Lisa were key in commenting on the user interface design, had a lot of suggestions and a lot of good criticisms. And Gail actually had been at Xerox before. So, she said, "I don't like pull-down menus. [I] think they're ugly, and they cover things up." So, you know, I'm trying to cut something, and the first thing that happens is the menu comes down, and covers the thing I'm trying to cut. No good.

And then Bill had this feature, that you could -- which he still has -- you could drag the mouse along the menu bar, and the menus would pop up one at a time. And we thought this is great: the user can see all of the available commands in one swoop. Well, she thought it was terrible, that you had to do that to see all the commands. I didn't understand, because in UCSD Pascal you had to hit keys for 10 minutes to see all the commands. So, I didn't quite get that.

She also didn't like the idea of dim highlighting. She thought we should just suppress the items altogether. So we had dissent in the group. And we always had dissent in the group. I just chose that because it's a memo that I had, illustrative of dozens of people who had dissent over every issue. Nothing specific about Gail. In particular, she said, "Last version the spec, I thought we only had one issue, which was when that little menu bar at the bottom was full, we didn't know what to do about it, and you should have just solved that, instead of moving it to another place, adding drop-downs, and all these other things. You over-designed to solve the problem. And besides, I don't like Z for Cut and X for Paste." She had her own proposal that was mnemonic.

September 22nd. It was now a month after I first said, "Bill, why don't you try at the top of the screen?" Bill had a very long night, and came back with the menu at the top of the screen, with command keys, check marks, everything -- he had everything. In one night he invented the entire mechanism of the menu bar at the top of the screen. To me it was just a place to put what he already had, but once

he realized it was there, he could do a lot of other things. And he implemented an amazing amount in one night.

So he wrote a memo to justify it, and said, "There are some problems with it. You have to go way further to reach the menu, that's a problem." But we decided, since there were command keys, and he made it very easy to have a single-stroke command key in this version, this all-night version, that okay, if you really wanted to go to that menu item a lot, you could hit the command key on the keyboard instead and you wouldn't move the mouse at all. So that solved that problem. The other problem was, that if there were a lot of windows open, you wouldn't really be sure which menu applies to which window, and sure enough, that's still true today. But, tradeoffs, you know. Why was it good? It was good because you got rid of all those ugly cases of sticking out menus, and also, the dialog box that came up when you had a command with parameters, could always go in the same place. And we had a problem before, with the dialog box coming up and covering up the menu, half the window, and so on. And so we could put it in a standard place under the menu bar.

So this was all the result of his long night of invention. And then, a couple more examples, we could put some global commands in the menu. Before, we didn't -- we hadn't figured out yet where to put the global commands. They didn't really go anywhere, because every menu was in a window, but now having a menu at the top, we could have global commands like shutting down or whatever. And then we could widen the title tabs, because the menu didn't take up room at the top of the window.

Now we're in February '81, so we've leapt ahead now several months, where all the programmers are busy implementing stuff. They're designing an operating system, telling us that we're holding up the project when they're still designing the operating system. I was in charge of the Applications Group, and we're the reason the project is behind schedule, even though they still were designing the operating system.

So I wrote a memo for other people who were going to help out with user testing. I made guidelines for user interviews. So, never argue with each other in front of the interviewee. After they leave, you can shoot each other, it doesn't matter, but not while the subject's there. Don't patronize the subject -- don't show off your knowledge. I wrote this because I would be bringing engineers into the room, and I'd sit them in the back. And the user would struggle to do something, and the engineer would run forward, jump out of the chair and run forward, grab the hand of the person, and say "It's easy. You just do this." [Laughter] And I'd go, "Naw, naw naw. You don't get it." So, we tried gags, we tried all kinds of things. [Laughter] Finally I said, "Look, I'll let you in the room if you observe these rules: Don't plant the answer. Take turns asking questions, if there are several of us there, so we get different -- we may each see different things. And mainly, try to

get them to do the talking. This is not a lecture by you, this is a test of them, and you want them to verbalize all they possibly can."

[The] typical thing that would happen is, we'd bring somebody in, to prove that the user interface thing they put in was absolutely right and we were totally wrong, and then the end user would usually prove we were both wrong: both the things were hard to use, and then we'd have to go back and redesign it. Or, half the time, the user would say, "Well, why doesn't it just work this way?" And we'd look at each other sheepishly, and fix it.

Now here was an interesting one, for those of you who do user testing. You don't -- if you don't name things, you don't just say, "What would you call this?" People paralyze, like "Oh, you're asking me to invent a name." They just get totally paralyzed, they can't think of any. But if you say something like "If someone were to bring you this, if you wanted someone to bring you this, what would you say? Bring me the" And then people would come up with something. It actually worked pretty well. People have similar techniques, I've heard, for other things.

April 3rd, '81. Well, at this point they were starting to get very nervous about the schedule. The operating system design was done, I think, and so we were holding things up. And they decided the problem was that these engineers, instead of just writing code like good sheep, were going and worrying about whether it was usable or not. So they decided to take all power away from us, and put all the decision making into a council. So who was on the council? [The] first person was the Division Comptroller, from Finance. Second one was the V.P. of Sales, who was always on airplanes. Third one was head of Publications for Lisa, which was a good idea. Fourth one was kind of the architect of the group; that was a good idea. And the fifth one was somebody in Product Marketing, who you saw had made comments before, that were pretty good ones. So, I was looking at this, going "Umm, okay, maybe we can live with this." But the main thing they did that was very good is, we instituted a process for what to do if you had a user interface issue. Don't spend eight hours arguing with each other. Write it up, make your points, get back to work. And then we have a process for dealing with it. But, the council would rule by majority vote. So, I think, "What a way to design a user interface, by majority vote of the Comptroller, the Sales V.P., okay." [Laughter]

But, we started. This motivated those of us who decided we had to do this the right way to actually run more user tests. Because we thought maybe we could overwhelm them with facts, and that will overcome this reliance on votes of committees. So, Chris Doerr, Wallace Judd -- I think Wallace maybe worked at Xerox before this, I can't remember -- these are two people that got involved with user testing the Filer, which we think of as the Finder today. And they ran some tests, and came up with all sorts of comments. I've underlined a couple here. Like there needs to be a way to select multiple icons at once. Well, they weren't icons yet even. Ways to manipulate multiple things at once.

But basically, after making some comments, they said, "You know, this is a good idea." Their own personal experience with testing was that you hired a testing firm, you spent a lot of money, three months later you came back with test results, and you did some things, and you did it again. You know, it took forever. Or, they would take a book and give it to a subject. Have the user read the manual, try out the manual, videotape the whole thing. And she said, "You know, this wasn't bad, just sitting down next to the user and having a dialog with the user, trying to get the user talking, and express what they're doing, is not a bad way to do it." So that was good, that we got the people doing the user testing to adopt this much simpler and cheaper paradigm.

We tape-recorded the sessions, and they found that useful. They went back and listened to the tapes. However, the tapes were hard to hear, because the people in the room were making noise. So they asked the observers to be quieter.

But one problem, she said, was that we're only testing Apple employees. We would get Apple employees the day they took the job, the day they started work. We'd get them in orientation. The person running orientation would say, "Well, after orientation you can go straight to work, or you can go be a user test subject, and, hey, you can get out of work for a few more hours," so we got a lot of people to come over and be user test subjects. Almost all of them had very little computer experience, because although Apple would have liked to hire people with computer experience in those days, there weren't very many.

Here's a typical schedule. We weren't trying to do two an hour any more. One person at a time. One each day. Nine o'clock, one o'clock, three o'clock, much more civilized. And we had a checklist that the tester went through, before each session: Have you reminded the subject of whatever -- I've no idea, whatever they're supposed to remember, that's on the list. Is there coffee available? Do you have a blank questionnaire ready for the person? Because we had them fill out a questionnaire at the end. Do you have the checklist ready? Have you rewound the tape? Have we rewound the tape for this, the talk today? Have you rebooted? Because it crashed if you didn't. Are the cue cards in the testing room, etcetera. After each session did you cover up the Lisa again, because the room had a window, and at night you could see in the window, and see the secret Lisa machine. Did you close the blinds, for a double measure? Etcetera, etcetera. So we had a lot of things to remind the tester about. That's a good idea if you ever run user tests, have a checklist.

Now Wallace, the guy who received the last memo, wrote this memo about some testing results. He said he spent about an hour a subject, which was pretty typical. That's the way I like to do testing myself. And he said, "Well, some functions still unclear; in particular, seven subjects out of" -- how many? It doesn't say, but not a lot, not very many -- "most of the subjects had trouble with Undo Last Command. They'd never seen such a thing in a program before, and they just didn't quite understand what it meant." Then there were other ones they

understood, but they didn't feel comfortable with somehow. One was Print a Copy, one was Mail a Copy. Well, I believe the reason was that we didn't have printers yet on the machine, and we didn't have e-mail. In fact, we never had e-mail. [Laughter] So, we're probably just showing these, and not really -- couldn't use them, so of course they didn't understand them.

Graphics editor user testing. This was LisaDraw, which became Mac Draw later, implemented by the same guy actually, Mark Cutter. And we wanted to test various things. Is this confusing, is that confusing? What happens if you try to use rulers at the same time as cross hairs? Is that too many things on the screen? Considering all the widgets that are there today in a Photo Shop or something, why were we worried? There was even worry about whether people could use a graphics program, because most people can't draw. So they wouldn't use a graphics program.

We also wanted to know the error rates. And so a lot of counting was being done, and looking at tapes to see how many times people made errors, and try to reduce the error rate. So we were trying to be sort of scientific, even though we were on this headlong schedule. You may have noticed that six months have already gone by and we haven't shipped. In fact, it's about a year at this point.

Okay, so June '81, Pat Marriott writes this memo about validating the prototypes, and Trip Hawkins at this point is at his wit's end. He says, "Aren't we done with all these decisions yet? Let's get on with it! Why are we still talking?"

Here are some of the issues. Should the arrows point apart, away from each other, or towards each other? Because, after all, the text moves in the direction that's opposite of the arrow direction, we could flip the arrows and make them point to each other. And I wrote a comment here, "Ahh, we can wait until September to test this. No hurry." I wasn't in synch with Product Marketing, who wanted all the decisions to be made, because I knew that changing this was a matter of about three minutes of programming. So what was the hurry? I'd rather work on harder issues, like getting the software to fit in memory, or something.

And there were some questions about clicking in folders -- oh, by the way, when you see folder here, we used to call windows folders. I forgot to mention that earlier; it might have been confusing here. Every window is a folder, even a document window was a folder. Not just the ones on the desktop. And we wanted to do some tests on that.

[Inaudible comments from audience.] Okay, October '81. We took Chris's advice to heart -- Chris Doerr's -- and we did outside testing. 24 people who did not work for Apple were done in a 1-way mirror room in Daly City. They didn't know what company it was that was showing it. A blind test. The psychologists had their day. We got to do this right. This was the thing that, there was no way we were

going to do this, too expensive, takes too much time, but by this point Product Marketing was going, "Well, maybe we'd better test this on real people."

We wanted to know whether they could read the manual, and install the hardware, and get the thing running, by themselves. And it turned out that it was 10 minutes, on average, that a person could, from the time they opened the box, until the time they get set up and running something. We thought that was pretty good. Those of you who were at Apple later know that on the Mac, it was even easier, at the beginning, but as the Mac got more and more developed, that number went up. And when that number started getting about an hour, Jean-Louis Gasse got extremely upset, and he ordered new user tests, and they ran user tests about that, and got the time back down to 10 minutes again. But there were some bad years in there. It was very hard to open a Mac box and figure how to set it up.

Two thirds of the users say they felt comfortable using the product as a result of the first experience. And about half of the people who were -- half of the third that were uncomfortable were people who happened to run into crashes and so on. Which probably affected -- they were trying to beat the clock on setting up the system. It was kind of frustrating.

And we had a questionnaire they had to fill out. This is the result of the questionnaire, so those numbers are how many people had that answer. And they didn't think it would be that difficult to set up. Before, they started ... [few words missing from tape] ... after they did it, they found that it was very easy.

All right. Now it's still 1981, I should mention that the Xerox Star was announced at NCC or something -- National Computer Conference, I think in June of '81. Some Lisa people flew out there, got a look at it. Bill Atkinson went, I believe, and Steve Jobs went, or maybe he sent Bill and didn't go himself, I can't remember. At this point we had a guy named Greg Stikeleather who had joined the group, working for Ellen Nold in Lisa Training. Some of you may know Greg, because he's more recently been an entrepreneur, sold a couple of companies. And Ellen is sending a memo to various people, copying Greg, on responsibility charting for user tests. Now, that user interface council we talked about before, with the majority vote, I don't know if they ever even had one meeting. We kind of overwhelmed them somehow. And by this time, a few months later, we had to have a new process. So the new process was that, based on some lecture we all went to by some management consultant, we were going to have people with authority, people with responsibility, people who are consulted, people who are informed. And basically, Product Management had all the authority, since Ellen worked for them [laughter], and Engineering and Training, which was Ellen's group, had all the responsibility. So we did all the work, and then Product Management would bless it and say that ... [portion missing from tape]

Okay. February 1982. A year and a half has gone by. Engineering user tests. This is a sample test. This is a write-up of a test. We had a standard form to write them up in, now we're getting real formal here. Not just, you know, run in to Steve Jobs after the test and tell him what happened. We were writing formal reports: what the person did, how they did, I'll skip over a little bit of this. The person had to draw an org chart, and here, not having a graphics program myself that had printing working yet, I was doing it in the word processor and drawing little org charts here. But they were really doing it with a drawing program.

And by the way, the thing that won over the Product Marketing people was when we gave them the drawing program. It was the first one to get working. I had Mark Cutter just try to get it really reliable, and don't worry about too many features. And then we gave all the Product Marketing people a Lisa, some temporary operating system, and the drawing program. That's it. It was a single-tasking operating system at that time. And they all started using it to make foils. You can do text, you can do boxes, you know, and they all did foils. They fell in love with the Lisa, and even started to understand the Lisa, and gradually the complaints started dying down, and they started trusting like maybe we knew what we were talking about. So that was a very good political move. I always made sure Mark gave them good stable versions of the Draw program. They'd draw a floor plan. And then at the end ... [few words missing from tape] ... had comments to make, and you could see that the comments were summarized, and people were basically very positive about this program. Sometimes there were complaints, and we addressed them.

This particular test was, I guess, run by Greg Stikeleather and me. He probably ran the test and I was in the room, or sometimes we switched off. He made a comment, which was that I-beam, which was the cursor that you used to select text, when the person would start typing, that little cursor was covering what they were typing. So he suggested that we get rid of it. Well, there was no mechanism to get rid of it. in the software, but he didn't know that. So it was good that he was there, and suggested it. And we then implemented that.

I pointed out that the stretch handles were a problem. Initially, what Mark had done was he studded the outside of the object, like every fourth pixel, with a little handle. So big objects had beads all the way around the outside. It was like a necklace. This really confused the people, and I suggested that we just have the four corners and the four sides, and that's all. And that is the way it ended up.

Of course, I'm going to show you all the good comments. We had a lot of stupid suggestions, too. And now, remember we had user test guidelines before. Greg wrote some new ones. Here, he was trying to address this to people who might have been used to formal user testing, and never experienced what he was calling development testing -- things you test after the fact, and things you test while you're still in development. So he made some guidelines, and they were

interestingly a little different from mine. A different level. And I think I didn't bring them. Just the fact that he did.

All right, getting near the end here. May, '82. A little less than two years since I arrived. I had just finished some tests, so Greg said we should run some development tests, so we did, we ran development tests. I taught 30 Apple employees, over a period of several weeks, several different applications, to see whether we were on the right track. By this time people were accepting this as a good approach, because it was a lot better to be informed by a user test than to just see who could talk the loudest in an argument, or who had the most political clout.

What I found was, the way we taught it made a lot of difference. You could take the same user interface and teach it a different way, and people would get real confused, or understand it, or make more mistakes, or fewer mistakes. And terminology made a difference also. So we then started a terminology project that Ellen Nold ran, which ended up, and I have many slides I didn't bring on that, and that ended up with the File menu, the Edit menu, et cetera, as you know today. And the various commands that were in them, choosing all the words for everything, that have pretty much survived into the Mac. I'd say 95 percent of them survived into the Mac, and most of them went into Windows also. So that was a very healthy terminology exercise. We did a lot of testing, and a lot of debating, a lot of linguistic analysis.

One thing that I say here, is that after all these tests, I discovered a lot of things that could be better, but we do have to ship a product, and it's been a long time since I've been at PARC, so okay, we'll ship a product. But couldn't we fix these in the next release? Well, little did I know there would only be one more release.

And then we had a process, again, where we were going to be very careful about prioritizing all the user interface changes, and measuring their schedule impact, and being sure they were worth it. [It] didn't have to be zero, necessarily, but it had to be worth the schedule impact.

I think by this time people were really realizing that our only differentiation was ease of use, and it better be really easy. And then I went through some problems. This thing went on, by the way, for about 30 pages, so I'm just going to show you one page. Cursor changes were not consistent, and there were cursors changing shape everywhere. The I-beam wasn't the only one, like, no matter where you went in the window, you got a different cursor shape, and I thought that's too many, let's simplify that.

We had double- and triple-clicks at that time, and some people -- it was the amount of time between the clicks mattered, and you couldn't control it. So I suggested we add a preference item, that people could control that time, which of course is what you do today in Apple's mouse control panel. And I also had some

comments about how you teach it, and in fact, when I looked at that whole memo, I'd say there were at least as many comments I made about how to teach it, as to what it is.

And then, months earlier, I think it was, maybe one month earlier, Greg Stikeleather had pointed out that the I-beam should disappear when you type. I don't know whether I forgot, or whatever, but I suggested the same thing again, or observed it, and I proposed we hide the cursor. And here in the margin is something I wrote later, saying it has been implemented.

If you're wondering about the cursor changing to a clock, or an hourglass, various other things, that was something that was actually at Xerox. Xerox had that at PARC. The way we did it was, we would wait a certain number of seconds, and if the operation wasn't done we would put up the hourglass. Trouble was, by that time the operation was probably done, or some of the applications forgot to put it up at all. So we came up with some clever algorithm for having it come up automatically, and disappear at the right time, and all that sort of stuff.

Now, here's the big change. The Star had come out a year earlier than this, and we were all told by management and ourselves, "Don't pay any attention to that. We've got a product to ship." But in everybody's heart they kind of felt like "You know, most of the Lisa's easier to use than the Star, but there's one thing about the Star that is really, really nice, which is those little icons on the desktop." There was icon envy that was very serious, and we weren't allowed to work on it. But around March of that year some of the guys got so troubled by it that they started working at home on weekends on an iconic filer, we called it. And at first it was a big secret, and they started letting people in on it a little bit at a time, and finally everybody found out about it, and there was a major uproar that people would dare to make changes at this late stage. But, they'd already implemented it. And so you couldn't complain about the schedule impact. [Laughter]

The person writing the manual thought it was kind of a good idea, so she kept saying it wouldn't affect her schedule [laughter], so nobody knew what to do. So I made a deal with Trip Hawkins, and Wayne Rosing, who was the engineering manager at that time: we'll run user tests, and let the users decide. So I ran tests, and you know what? It turned out that the interface we had, which I won't bother to show you, but it was kind of more like Find File -- that's the closest thing I can think of -- was actually equally good in terms of ease of learning, equally easy to use, same error rate, very low, people understood both, there was no difference. And we already had one, so why did we change? The last question I asked everybody is, which did you like better? And they all liked the picture one better. And so we made a decision that had nothing to do with ease of use, nothing to do with ease of learning, nothing to do with error rate. It wasn't a human factors decision at all, in the traditional sense. It was a decision based on what customers liked.

And there's a whole history I won't go into here about the icons. They didn't actually come from the Star, they came from something IBM did. I'm not going to get into that. Long story. You can read the proceedings of Apple vs. Microsoft lawsuit or something to find that out. [Laughter]

Okay, so this is my last slide here. This was a memo that included on it Chris Espinosa, who was in the Macintosh group at that time. And the Macintosh people, who had been doing a totally different user interface, at some point decided they really ought to do some subset of the Lisa interface instead. And so they wanted to get a lot of ideas from us, as they designed their user interface. We started copying the Mac people on it. And with that, September '82, I'll turn it over to Chris.

Q. You know back in the days when the arrows were the single scroll bar, did it take a second mouse button to move it in the other direction?

LARRY TESLER: No, you used one for both. Why don't we, if you want, we can talk about that after the talk. I forgot that I had a videotape here -- is that monitor turned on? What I've got here is just 30 seconds, because the tape is really bad, of one of the very first user tests, in July of 1980. And the one I have here, Bill and I took turns, and this one I'll be showing you is Bill with a subject. You can't see the subject. Most of the tape, all I can see is the out of focus back of Bill's head, but for about half a minute we can actually see the screen of the Lisa. So it's really quick, you'd better get a good look.

[PLAYS THE VIDEOTAPE -- NOT TRANSCRIBED]

See, here's that menu at the bottom of the window, this is really early. And there's that highlighted selection the person made with the mouse. And he just explained something very simple, and she didn't understand it, so he's starting all over. And now we get to see his hair, so I think we'll stop. Okay, so there's nothing much to get out of that, but I did promise to show some early Lisa user tests. Now I've done it.

Given the time, I think we're going to save questions to the end, and let Chris take off on his. Great. Thanks very much. [Applause]

CHRIS ESPINOSA:

As Larry said, the work that we did on the Macintosh human interface dovetailed with the end of the very fruitful period of Lisa human interface design. I cannot believe how much work they went through in a short period of months in the summer of 1980, to put together what we still see today in the human interface of the Macintosh, and the human interface of Windows. What was really interesting to me was that when Larry first proposed this talk to educate some new people at Apple, from some companies we'd acquired, on how the user interface came to

be, that a lot of the people he was referring to, a lot of the people who are co-copied on some of the documents, a lot of the work that they drew on, really had nothing to do with the graphic user interface, but had to do with a long standing user interface community at Apple that had evolved in a totally separate organization for a totally separate strain, and that was the Apple II and the Apple III human interface effort that was going on in the Personal Computer Systems Division, which was by that time physically remote down in the big black triangle shaped building at the intersection of 280 and Lawrence Expressway, called the Triangle Building or by then the Bermuda Triangle, because it's where the Apple II group had been moved to die. [Laughter]

The thing you have to understand about the crucial difference between the Lisa-Macintosh human interface effort and what was going on in the Apple II group was not the difference between graphic human interface and text-based human interface, though that was an important part of it. The crucial difference was that in the Lisa and Macintosh, the engineering groups were completely integrated in, and the driving forces behind, making the computer easier to use, whereas in the Apple II the evolution of that had been that the impetus for making the computer easier to use, and which programs were selected to make that, had come from the Publications Department, the Training Department, and the department doing demonstrations for in-store retail sales, and I was a member of that group. Our effort was primarily to mask the complexity of the real underlying software that we were already shipping, by putting sugared user interfaces on top of the real interfaces. And by putting training programs on top of the real computer programs. This was -- remember the Apple computer by 1980, boy, we'd been around for three years. We had a long history behind us in the personal computer industry. And we had a lot of accumulated legacy software that we couldn't just haphazardly change, we really had to work on it.

One of the documents that I have here is a document; I've got it on video here, although I don't think it will be that easy to read.

VIDEOTAPE: ... manual is consistent and provide a consistent and familiar and productive experience to readers of manuals. Bruce Tognazzini took the concept of the ... [sound fades out]

This is that document, and this is the change history of the Apple II human interface guidelines, that show the first draft Bruce Tognazzini claims to have written in September of 1978. A second draft in March of '79, third draft in mid-1980, fourth draft with Steve Smith in early 1981, and then it got into the Macintosh group in 1982. I started working on it in late '82, and that's when I incorporated many of the things from the Lisa efforts. And then it moved back over into the Apple II group, to cover the Apple II graphic user interface. But it originally started as an Apple II text human interface document. And it came originally from Bruce's efforts with J. D. Eisenberg, writing an application called

Apple Presents Apple, which was the main in-store, or first user demonstration program for the Apple II.

Bruce started out -- well, I don't know where Bruce started out, but when we encountered Bruce Tognazzini in his long and checkered career, he was the proprietor of a computer store in San Francisco, which is where the Apple Corps of San Francisco used to meet. And that's how he got into the Apple community, by selling Apple computers. So he had front line experience in what happened when a person encountered a personal computer for the first time. When he came to Apple he started writing demo programs with J. D. Eisenberg. And one of the things that they did was, they tested on real users in real situations, where you bring somebody who's never seen a computer before, much like the Lisa group was testing people off the street, or new employees, put them in front of this program and gave them no advice whatsoever.

I'm going to read -- Bruce tends to philosophize a lot in this, so it's not going to be as useful to put up slides, but I'm going to read to you some of his experiences from the -- here we go. From user testing: "Our testing method is as follows. We set up a room with five to six computer systems. We invite groups of five to six users at a time to try out the systems, often without knowing that it is the software, rather than the system, we are testing. We have two of the designers in the room: any less, and they miss a lot of what's going on; any more, the users feel as though there's always someone breathing down their neck. The initial ground rules are that no questions will be answered, as by the time the formal testing begins, we can supply a draft of the manual. Usually by the second group, some glaring defects in the interface have shown up, and we have to give them help getting past the stumbling blocks. 95 percent of the stumbling blocks found are found by watching the body language of the users. Watch for squinting eyes, hunched shoulders, shaking heads, and deep, heartfelt sighs." [Laughter] "When a user hits a snag, he will assume it is his fault. He will not report it. He will hide it. Make notes of each problem and where it occurred. Question the users at the end of the session to explore why the problem occurred. You will often be surprised what the user thought the program was doing at the time he got lost."

"Herein follows a true anecdote which illustrates how difficult the most simple human interface issue can be, and why thorough testing on real people is so important. As we tune in" -- it's so easy to entertain a crowd just by reading something Bruce wrote [laughter] -- "as we tune in, the authors of the software, both of whom pride themselves on clever interface design, have anguished for hours over difficult passages in their program. It was to turn out, their guesses were quite accurate in said difficult passages. It was the simplest question of all, the cause of the problem. In Apple Presents the Apple II E, the training program for teaching fundamentals using the new Apple II E, find out if the user is working with a color monitor.

"User profile: new owner, customer in computer store, or member of a class learning how to use an Apple computer. Test user profile: customers in computer store, non computers [users?] in classroom environment, friends and relatives. First design: a color graphic is displayed on the screen. Prompt: are you using a color TV on the Apple? Anticipated problem: those who were using a monochrome monitor in a classroom or a computer store situation wouldn't know whether the monitor was black and white, or was color with the color turned off. First attempt: a color graphic's displayed, prompt: is the picture above in color? 25 percent failure rate." [Laughter] "Reason: as anticipated, but incorrectly overcome, those seeing black and white thought their color might be turned down. They didn't answer the question wrong, they turned around and asked one of the authors whether the monitor in question was color or not. A decision was made: the authors could not be supplied with the disk." [Laughter]

"Second attempt: a smaller graphic, with large letter words in their own vivid colors, was substituted: green, blue, orange, magenta, which were the only colors the Apple II can display. Prompt: are the words above in color? Failure rate: color TV users, none; black and white monitor users, none; green screen monitor users, 100 percent." [Laughter] "This is why it pays to have a variety of configurations in your test suite. Third attempt: the graphic remained the same, prompt: are the words above in more than one color?" [Laughter] "Failure rate: color TV users, none; black and white monitor users, 16 percent; green screen monitor users, 50 percent. Reason: the black and white monitor users who answered incorrectly admitted they did so on purpose." [Laughter] "50 percent of the green screen folk considered that they were looking at both black and green, two colors, and answered the question accordingly." [Laughter]

"Fourth attempt: same display of graphic and color text. Prompt: are the words above in several different colors. Are the words above in SEVERAL different colors? Failure rate: color TV users, none; black and white monitor users, 20 percent; green screen monitor users, 23 percent. By this time the authors were prepared to supply everyone who bought an Apple with a free color monitor just so we would not have to ask this question." [Laughter] "It turns out about 20 percent of the people were not really reading the question, and were responding to 'Are the words above several different colors?' 'Well, yes, of course, green, violet, red, there are several different colors, let's move on!'" [Laughter]

"Fifth attempt: same display of graphic and color text. Prompt: do the words above appear in several different colors? Failure rate: none. In case it appears the authors were simply dull fellows, be it known that this was a fully interactive training program, in excess of 100K of code," -- remember, it was a 128K machine, so ... -- "and this was the only interface issue that required more than one correction. It clearly exemplifies how even the most careful designers can totally miss when guessing how users are going to respond." So, reading from Bruce Tognazzini.

The Apple II project had started out -- well, the Apple II wasn't really a project, it was a collection of projects. We started out with a number of pieces of software that had our cherished angle bracket or square bracket prompts. By the time we got around to the Apple II E and the Apple III, we'd developed a more or less text windowing interface, called File Card Metaphor, where we used special characters in the character generator ROM to do something like the File Card interface in the Lisa, with -- pick one, two, three, four, five, six menus, and you used the escape key to back out through a hierarchy, and press numbers 1, 2, 3, 4, 5, 6 to advance to the next level. It was a hierarchical-spatial text-driven, keyboard-driven user interface. It didn't use the mouse. It was very successful for what it did. We ensured consistency across many applications, all the applications that Apple implemented as well as third party applications. For example, if anyone was familiar with Quark Catalyst -- Quark, they're the people who make Quark Express now, they made the original Hard Disk Manager for the Apple II and the Apple III, and it used the same interface. It had consistent use; in the keyboard keys, the Open-Apple always meant the same thing in different applications. And so even though it was a text interface, it had many of the underpinnings of the consistency of the graphic user interface of the Macintosh. And it was driven by the exigencies of User Training, which was my mother's group, because the less time we had to spend on user training, the cheaper and faster it was, and if the software were consistent, the training went more easily.

It was driven by the exigencies of Publications, which I was running, and one of the problems we faced in Publications was that having not invented desktop publishing yet, we had six weeks minimum from film to printers, which was usually the time that was reserved to write or finish the software. We often had to be finished with the manuals several months before the product shipped, and the software wasn't finished at that point, so in order to have manuals that accurately reflected what the software was going to do, we had to have the programmers tell us what the user interface was going to be, and then in going through that with the programmers, having them do it the same way other applications did it gave us a more full idea for "okay, it's going to do this, it's going to do this, it's going to do that," if they followed some standards or some guidelines. In fact, what Bruce did before writing the Apple II human interface guidelines, and what the origin of this document was, was in fact a Publications Style Guide, which was how we wrote manuals and how we explained portions of the human interface, and explained portions of how you used the computer. And by using consistent explanations, that drove consistent design in some of the applications.

So that was in the origins of user testing and of consistency of user interface design in the Apple II group. As I said, it moved into the Lisa group, starting with the separate angle which Larry talked about, coming from Xerox PARC. And the two basically fused in the Macintosh group, where we had the Lisa -- the benefits of the years of Lisa user testing experience and the massive amounts of design that went into the graphic user interface, plus a lot of the democratic, grassroots issues that had come up through the Apple II. Remember, Lisa was only dealing

with seven applications. That was all you got on a Lisa. The Macintosh was designed like the Apple II with something where you had many, many applications from third party developers. And so we took the Lisa user interface, grafted it on to the guidelines that had come out of the Apple II Style Guide and Apple II guidelines efforts, and published that for third party software developers to use as style guides for their own applications.

I remember very, very, very clearly that one of the massive controversies around the development for the Macintosh, circa 1982-1983, was developers would come up to us and say, "You know, if you make the user interface consistent, and if you put all that software in ROM that makes it -- you know, if you make it hard to write to the screen directly, so that we have to use your user interface software to talk to the user, how are we ever going to make our applications unique, and stand out, and be different from each other in the marketplace?" They found a way, I'm very happy, but the human interface and the idea about consistent human interface, much less a graphic one, was very, very controversial among software developers. And only by having the user testing under our belt to show them that hey, this is going to add significant value to your product, by making your product easier to learn and easier to use, especially for people who already know another application, were we able to really convince them that a consistent graphic user interface was a good thing to have.

I want to fast-forward a little, skip through the early years of the Macintosh and go to a tape that I dug up from the Apple library, of some user interface testing that shows how amusing user interface testing can be, and the kinds of situations in far-flung places that members of our glorious user interface group went, to find the truth about what real users would do. This is a tape of a couple of our user interface designers, Lauri Vertelney and one other person, I may come up with her name. [Inaudible comment] I don't think it was Gitta. Traveled to France to a parts dealership for Renault automobiles, who were deploying a new automotive parts catalog based on HyperCard, HyperCard being a free-form tool kit for designing new software interfaces. [It] presented an entire raft of new problems in human interface design. They actually went to France to test auto mechanics in their own environment. I'm going to play that tape for you now.

[Q, while videotape is being set up:] Was the name Kate Gomoll, is that right? Do you know the name?

CE: It might have been Kate Gomoll, yeah. Thank you, Annette. Was that Annette? [Inaudible reply] Oh, it's Joy. Hi, Joy.

[VIDEOTAPE PLAYS]

"The first part of the user test is to test users attempting to fill out the order form and the estimate form, using the current paper methods. These are the order and estimate forms that the mechanics currently use. The [inaudible] part catalog is

used to look up parts information. And the repairmen will have instructions on how to [inaudible] car. Kate's going to test the procedure and tasks to be performed by the mechanic. All the instructions were interpreted for the mechanic because Kate doesn't speak French. [Several sentences in French.] If you have trouble with some of the tasks, it's the system's fault, not yours. [More French sentences.] [Inaudible] Kate suggests that we have us evaluate the results later, [inaudible] after the test session. Seeing the mechanic work with the paper helps us to understand how he prefers to get information. For example, he stacks the forms on top of the catalog, and flips the pages while he's searching for information. Research techniques helped us to submit one recommendation for searching with the electronic version [inaudible]. Once we have finished testing the current form [inaudible] procedure, we ask the mechanic to solve the same problem with the documentation for the electronic system.

"At this point in our test the user has tried for over two minutes [inaudible] to start the system. Kate finally instructs the user he must click on the tiny icon below the Renault logo to start the system. The interpreter wants to know if he should explain the popup menu that [inaudible] below the startup icon, and Kate says no. When he clicks on the icon he knows nothing about how to use popup menus, and initiate the dialog box, which explains to him he needs to hold down on the icon. Of course, the mechanic's never seen a dialog box before, and is confused, not only about the message inside of it, but what to do with it. Finally he realizes he must click on OK to get rid of it. [Inaudible] he tried to stop the application and click on any of the buttons in the menus in the left of the screen. Unfortunately, none of the menu items worked at this point, because he had not filled out the appropriate vehicle identification, to actually start a work session. He tries the small icon in the center of the screen [inaudible] the dialog boxes telling him to hold down on the icon. This time he doesn't see that he must click on the OK boxes, but he's clicking around the screen [inaudible] Finally Kate instructs him to hold down the button in order to start the program [inaudible].... hold down the OK button in the dialog box." [Laughter]

"We tried testing the dialog [inaudible], one more time. But, [inaudible] the dialog box. [Inaudible] the OK button [inaudible] the popup screens are not going to work."

[END OF VIDEOTAPE]

So, that just, I think, is an illustration of two things. One, that bad design knows no bounds. And two, that you never get a user interface right, and that user testing, from the earliest times when you have a totally unsaturated market -- people who have never seen personal computers before, much less mice, keyboards, screens, things like that -- to today, when we assume that anybody who's ever been in western society has had to have encountered a personal computer at some point, and knows the basics of pointing, clicking, dragging, pressing, what a file is, what a document is, how to print, why to print, what a

floppy disk is. I still find it surprising that we don't talk about sectors and cylinders any more on disks, which was one of the main things we had to explain in 1978, 1979, because there was some reason you needed to know that, I forget why. [Inaudible comment] Yes. But that some things never go away, and that having to test new designs, and test your assumptions of what your audience knows, is pretty much eternal.

So, at that point, having run pretty late, I think Larry and I will take a few questions, and I think that we're going to take free liberty to defer to several people in the audience who are noted in Apple human interface history. I see Joy Mountford and Annette Wagner back there, who've been instrumental in human interface groups in the past, and if there are any others, and if you have strong opinions, disagree with us, have more information than we're able to supply, please chime in. And Larry, do you want to pass the mike around, so people can ask questions in the mike?

LARRY TESLER: [inaudible] to repeat the question.

Q: Larry, I wonder if you could comment on the story that's often told, that Apple ripped off the Xerox PARC work. I know Jef Raskin has refuted or denied that, but it's such a popular story, could you go on record as giving your perspective?

LARRY TESLER: Yeah. Well, I actually showed, at the beginning of the talk, a picture of the Smalltalk interface, to see that, in fact, the early user interface designs were very close to that. The look of the windows, in particular, and the icons and the desktop. But it evolved over time, and the menus moved to the top of the screen, and so on. If you actually look at either a Xerox Star or a Smalltalk screen, and then look at a Lisa or a Mac, I think you'll see a lot of differences. What was taken was the concepts. The concepts that there ought to be overlapping windows, that there ought to be direct manipulation, and so on. But the Xerox Star, for example, didn't have the concept of dragging things. And it had a two-button mouse, you'd click one place and then click another place. It didn't have a menu bar. In fact there was only one menu, that was added late in the development. Some things were similar: there was a horizontal and a vertical scroll bar. So certainly the concepts came largely from there, but as you see, some of the concepts came from other things: from HP systems, UCSD, from Apple II, from some work that was done at IBM that I didn't talk about today, that was published around the late '70's. So, there were a lot of origins for it. I think the Xerox work had a lot to do with it, but usually when the -- if you sat down to try to use a Star or Smalltalk you would find that there were I think more differences than similarities in what you would actually see and do.

Q: [Inaudible]

LARRY TESLER: The question is, the user testing we showed for the Lisa had to do with novice users, and the question was whether we tested for how people

acculturated over time. Well, the system, remember, from the very first page of the user interface guidelines Bill wrote was targeted at novice users. That was the target, which was about everybody in those days. And we did have some theories about how people would use command keys as they got more expert and so on, but we didn't run user tests. One reason we didn't was because it would have taken a long time to get people up to speed. So what we did was, we just used kind of ourselves. I don't mean the engineers, but people who worked in the Lisa division, who had Lisas, like people in Product Marketing. They became expert after a while, and they would start telling us that things were too slow, or too hard, or whatever. And that's how we got that kind of feedback, just from people inside. I think it's more relevant today to check for those sort of things than it was back then, when everybody was -- you know, 95 percent of the people were novices.

CHRIS ESPINOSA: One of the constant tensions, and this was true on the Macintosh as well, especially because we had the benefit of the Lisa work before us, and some of the Lisa machines, the constant tension is that people who are designing are almost always by definition expert users, and will often overlook that what they think is obvious is not at all obvious to a novice user. And so there's constantly the dilemma which you've seen historically in Mac system software, that the expert users want to put in the features they want to use, but the people who want to keep the system pure for the novices want to resist those, and if you're lucky you get a system that is easy to approach for the novice, and gradually unfolds itself for the expert. And if you're unlucky you get a lukewarm mediocrity between the two, where it's a little too complex for the beginning user to understand, but still not nearly powerful enough for the expert user. And some of our designs, and some of the industry designs, have been gradually unfolding, and other have been just plain mediocre.

Q: I have an observation and then a question. Larry and I have known each other for years, from Xerox days. And it's interesting to me that, for one thing, you folks weren't aware of the similarity to the Wordstar commands that already existed, in the Wordstar editor, for Copy, Cut, Paste kind of things. But also I think the word "assumption" is key and what you just said is very correct. People assume as they're developing something that they know the best way to develop it, rather than go outside and get other information. One of the things that I've seen as a software person in the past, having to do work for people for money, so that they expect us to have something done by a certain time, not for a company that I'm working for but as a consultant, is that they really do expect the thing to work the way their people want it to work. And there you really have to do a design review, and all sorts of things with those people first before you get to the right setup, before you can actually start writing code. And one observation about the Lisa that was very interesting, when I first saw it -- I first saw it, I think, at the West Coast Computer Faire in San Francisco. Steve Jobs was there, I don't know who else from Apple was there, other than Marketing staff. But I'd gone there with a friend from Xerox. And we looked at the Star system. And we'd actually looked at

the Fortune system, which was the first 16-bit personal computer that was available. And they had a very fancy display, and so forth, and everybody was looking at it; it was beautifully designed, and everything. Everything else that it ran CPM and UNIX, a version of sort of UNIX. There was one test in the software work that we had done, for real customers, who really wanted reliable software, that we'd accidentally come upon which was called the "keyboard mush." You take your hand and you run it across the keyboard a few times, and you see if your software is still running. [Laughs] And I told this friend of mine from Xerox about it. So he walks over to the Fortune system, on the big podium up there, with the pretty girls standing by it, and he tries it. And the thing dies. So we kind of sneak away, and we go over to the Apple booth, and there's Steve and the other folks there. And we decided we weren't going to try it on a Lisa. So in reaching for a brochure, though, I accidentally touched the upper right portion of the Lisa keyboard. They were networked at the show, for a demonstration of networking. It brought down that Lisa. The guy helped me try to reboot it. It wouldn't come back. It also brought down all the other machines at the booth, on the network. And the only thing we finally could do was to power cycle the whole system. [Laughs] So I just wanted to bring that out. I just wanted to bring out the basic nature of testing, which involved looking for the unforeseen.

LARRY TESLER: Ah., looking for the unforeseen. Right. Good point. In the interests of time, we're going to keep the comments shorter, though.

Q: Do you see any relationship between [inaudible] and the graphical user interface [inaudible]?

LARRY TESLER: Yes. The question is whether there's some correspondence between the object oriented programming metaphor and the user interface metaphor. And yes, there is. In fact, in the Smalltalk user interface, which was one of the first of this type, and the Smalltalk language, which was one of the first of the object oriented type, the connection was very explicit: that you had objects that had generic commands that could operate upon them. And that was something that existed both in the language and in the user interface. Which was done even more so in the Star. But the programmers who were doing these user interfaces definitely had that in mind, as a correspondence.

Q: [Inaudible] a comment on the first [inaudible] counterexamples [inaudible] user interfaces before their time. Graphical user interfaces that had nothing to do with object oriented programming languages.

LARRY TESLER: Oh, yeah, good user interfaces don't have to, I'm just saying that the user interfaces that we were developing, we had that in mind. But that doesn't mean that a good user interface has to be object oriented, no.

Q: You showed two years of memos that you had written and received. Having come from an environment with the Alto, Bravo, Gypsy and [inaudible], what did you write all those on?

LARRY TESLER: What did I use to write the memos? Probably Apple Writer. let's see, early on we were using -- maybe Apple Writer on the Apple II, or maybe UCSD Pascal program editor being abused on an Apple II. I never had an Apple III.

CHRIS ESPINOSA: You're right, you were probably using either Apple Writer on an Apple II, or sophisticated people used the UCSD Pascal editor and output it using an NROFF-like program called Apple Script [LARRY TESLER: Right] and we output it to Qume printers when we could afford the time and noise, and to dot matrix printers, 80-column dot matrix printers, and you had a mix of those two. Before we got the Apple II's and Apple III's up, Jef Raskin had imbued us with Polymorphic Poly 88's, using their editing system, and we did all of our editing on Poly 88's, Soroc terminals.

Q. Back in the, probably 1983-1984 time frame, Digital Research had a graphic interface called GEM. That sort of got buried in litigation of some sort. Can you comment on that whole story from Apple's side, and speculate on what might have happened if Digital Research didn't [inaudible] part?

LARRY TESLER: Well, I don't know about successfully competing in that part. But yeah, it was extremely close to the Mac user interface, the GEM user interface by Digital Research. And Apple sent some kind of complaint letter, and there were some discussions between the companies. But then Microsoft and Hewlett Packard started doing Windows and New Wave, and that was a much bigger issue, so Apple kind of stopped bothering Digital Research and instead got involved in a big law suit with HP and Microsoft. Digital Research at the time was kind of a struggling company, and Apple didn't want to basically pick on a little company. Instead, we picked on two companies that were too big [laughter] instead of picking on one that was too little. Bernard?

Q: Would you like to comment on the fact that both Lisa and the Xerox product [inaudible] ?

LARRY TESLER: Yeah. Like the Xerox Star? And the Lisa -- the idea was, in both groups, was that people would only use a few applications. It's kind of true, most people only do use a few applications. But the conclusion that, therefore, it wasn't necessary to have third party applications was kind of ridiculous. Actually in the Lisa, it turned out we did have a place for third party applications. We thought for certain vertical markets, there would be third party applications and we'd have accounting software and so on. But even then, we were thinking very small scale, like there would be one page layout application, and there would be one accounting application, and a couple of vertical market things. If we ever had

20 applications, or 30, that would have been a very large number. Very different from the Mac, which from the very beginning was going to try to do the same thing as the Apple II, and what the IBM PC did, which was go for lots of third party applications. I think one reason the Lisa was designed as it was, was to try to justify why there was this project going on in the company, when we already had an Apple II and an Apple III. Why did we need this other thing? Well, the Apple II and Apple III were going to have all this third party software, but the Lisa won't. It was kind of a differentiator. It was a kind of a closed system that was supposedly a plus. It turned out to be a major minus, of course.

CHRIS ESPINOSA: Well it was -- that was not necessarily unique to Apple at that point. Remember one of the leading software companies of 1979 -1980 was Personal Software, which became Visicorp. And their flagship product line at that point was a failed product line called VisiOn, which was an attempt to expand VisiCalc into a comprehensive suite of interacting applications -- a word processor, a data base. Wordstar at that point was also trying to expand. Everybody was trying to do what Microsoft has successfully done with Microsoft Office, which is turn the thin end of the wedge of getting one successful application, which Microsoft had with Excel on the Macintosh, and turn that into a major suite that delivers all of the application computing needs of the given set of users. Microsoft has done it successfully, and one could argue that the default computing platform right now is Windows with Microsoft Office, and not very much else. So I don't think that that necessarily was a bad idea. It may have been many years ahead of its time. But Microsoft was selling the Microsoft Office for Macintosh in 1985. I still have a box on my shelf where they supplied the word processor, the spreadsheet, the filing program and the terminal program in one box, for one price, for the Macintosh. So, office suites certainly were not a dumb idea of 1981; it's something that has really monopolized the industry.

LARRY TESLER: Yeah, I think that part's true. The Lisa was really one of the first suites, in addition to the ones that Personal Software did. It was the first, probably, multitasking personal computer. But basically ahead of its time and missing the main point, which was, you needed a low cost platform that had reasonable performance and that had lots of third party applications.

Q: You were testing totally naive users without any instructions, by and large. And yet, in the intervening years, acculturation has taken place, and I wonder if that really helps us [inaudible].

CHRIS ESPINOSA: The point is that we were testing in the early years for totally naive users, and that over the years acculturation has taken place, as I talked about. More people approach computers with less fear, with more pre-knowledge of what they're doing. Is that your general point?

Q: Yes, and that pre-knowledge came from a process of about [inaudible].

CHRIS ESPINOSA: And the process came from a decade or more of experimentation. Well, I'll tell you about an interesting user experiment that I did. In 1982, I took a[n] early prototype of the Macintosh, and it was then in a Plexiglas case, just cut panels of Plexiglas, filed and glued square -- we didn't have any of the injection molded cases yet -- with the guts inside. It was running an early version of Mac Paint, which was called Mac Sketch at that point. And it was a turnkey application: you put the disk in and you booted it up and it went straight to that application, because there was no Finder at that point. Which was actually the paradigm of most other machines, which was turnkey boot: you put the floppy in and you booted it and you got into that one application. And I took it over to my girlfriend's parents' house for Thanksgiving, where she had a three year old nephew. A three year old nephew, and this was in 1981. Video games weren't even that -- hadn't even gotten down to three year olds by 1981. That boy spent four hours drawing with Mac Sketch. He learned it in no more than 15 or 20 seconds, just, you press the button and draw things on the screen. And he was drawing things with an early version of Mac Paint, with virtually no tutoring and almost no language at all. With no acculturation at all. So, there were some things that we got basically right, about "human" human factors. If you can appeal to a three year old, and get a three year old with no language skills to understand how to do something productive in no time at all, that's not a cultural statement. You've done something right in design. And that's when I knew we were really on to something.

Q: ... the acculturation has no affect?

CHRIS ESPINOSA: I don't think it's had no affect. I think what acculturation has done is, it's made us soft. We don't address complexity any more the way we -- we sweated over complexity, in 1981. We were deathly afraid of complexity. We take it for granted now. You look at Frye's. It's scary, the kind of things we get away with.

LARRY TESLER: I think it has made some things hard to change. When Microsoft announced they were going to go to single clicks instead of double clicks on the desktop, some journalist said, "See, they're leaving Apple in the dust. You're still on two clicks, and they're moving to one." I said, "Our users have been doing this for 12 years. There's no way they're going to change. We're never going to change it. Maybe the Microsoft users, who have only been doing this for a couple of years can change." But as it turned out, even a couple of years using Windows 95 was too much, and Microsoft had to back off on making users use single clicks, in Windows 97, because even two years was too much. People won't change.

We're out of time now, but there are some people here who were involved in Lisa user interface work, all the way from the Apple II, the Lisa, and the Mac. I thought maybe they could stand up. Annette Wagner specially, who worked on the original Lisa look, and graphics and fonts and everything. And Joy, and Bob --

why don't you all stand up, all you Apple interface people. I think some of the users out here might appreciate you. Bob Glass, still here? [Applause]

LEN SHUSTEK: Thank you both for a great talk. I hope you'll join us over in the warehouse, just down the street to the right, and Larry and Chris, will you both be there to answer questions? Great. Thank you for coming. See you next time. [Applause]

[END OF VIDEOTAPE]

Transcribed by John Amos, a volunteer for The Computer Museum History Center
San Antonio, Texas February, 1999