
# Oral History of Taylor Pohlman

Interviewed by:
David C. Brock

Recorded: March 14, 2016
Mountain View, CA

CHM Reference number: X7738.2016

**David C. Brock:** Well Taylor, thank you again for joining us and I thought we could begin at the beginning and just ask you about when and where you were born.

**Taylor Pohlman:** Well born as they say at a very early age, actually back east in Fredericksburg [Virginia] and grew up around the Washington D.C. area. I went to high school there and when it came time to go to college, launched off to UT in Austin. I was convinced that nuclear fusion was the future and it was gonna happen. This was in-- what would it be, like early '60s right and it was gonna happen they said in about 20 years, which they say it's still gonna happen in about 20 years. But UT at the time had imported a bunch of guys from Europe and they were working on a Tokamak I think it's called. The torus -- twisted torus -- and so I was gonna be a physics major and I was gonna solve the problem and went to Texas and found out that I was not born to be a physicist. I couldn't picture myself in a white lab coat. I think part of it was probably when I joined a fraternity and discovered that there were other things besides science. But the net is I wound up in math because that was the simplest major to graduate with, given all the courses I'd taken as a freshman and a sophomore and that's what actually, historically, led me- led me into computers, sort of.

**Brock:** Did you have some experiences in high school that you think were important to the development of your interest in physics, in fusion, in this area?

**Pohlman:** Physics and science I'd always been almost self-taught since I was a little kid. About the only foundation course that I took in high school, besides mathematics, that helped with computers was I actually took typing. Which was more about the fact that there were very few boys in typing and a lot of girls. But <laughs> be that as it may but I was a touch typist by the time I got to college which was pretty rare in those days actually. I actually made some money in college typing papers for people because most guys at that point hadn't touched a type writer. So that meant key punching came a little bit more natural to me and unfortunately that was about the only computer that was available, certainly to undergrads at that point. UT had a pretty leading computer science department but I was not in computer science I was actually more in pure math. So that was really more a chalk-filled environment as opposed to electronic-technology-filled environment. But the net though was I took a couple of computer courses when I was, I guess I wanna say junior or senior year. First was a pretty simple Fortran course, the machine was the CDC 1604 and I think I told you the story. It had a tape compiler so you fed punch-- well you actually didn't punch cards in the machine, you handed them to a guy who then put them into the card reader and you could watch through the window though, because they had a big glass window and the compiler would pull routines off the tape. It'd tick back and forth and so forth. But all the error messages were at the end of the tape. So you always knew when you were having bad luck because the tape would <makes sound> like that and would pick up the error message and you kind of walk away from the window and wait for the bad news. The-- actually I kind of hated it, one was it took at least two days to get that round trip feedback. So you'd punch your cards up, put your deck together, find out you'd made a mistake, have to go back, make the fix, turn it around. With luck you'd get your output by the end of the week and it just didn't feel very tangible. Then I took a course in machine language and I have no idea why I did that and I really didn't like that. It's even more tedious than the Fortran and so I bugged out of school with kind of an image of computers as this kind of boring, long, bad turn around, not very much

  

feedback and so forth and didn't really get back into computers at all until I started teaching math in high school.

**Brock:** Was it a common thing for your fellow students studying mathematics, math majors, to also be taking some computer courses?

**Pohlman:** No, not the theory guys, the engineering guys typically did, but UT in those days in math anyway had pretty much your-- as long as you took 24 upper division hours you could do anything else. So I actually minored in English and history and dabbled around with a bunch of other courses. Including computer just because I just wanted to see what was going on and I had no idea at that point of what I was gonna do. Now when I did get out of school -- and that would've been '69, '70 timeframe -- there were starting to be some real jobs in computers. The economy was pretty bad so I talked to RCA and a couple of other people but didn't really see anything I liked, and I'm not sure they liked me either given my attitude toward computers, and so I wound up teaching. Part of that was, to be perfectly frank, for deferment because that was, '69, '70, was the Vietnam era of course, and so I got a teaching job and it was at South Oak Cliff high school in Dallas where I was called a floating teacher.

So I had very low seniority so I didn't actually have a class room. I would, they would-- okay if it was second period I was in this room, third period I was in that room. Well, one of the rooms that belonged to the senior math guy in the department had a teletype computer terminal in the back of the class and it was tied into the old GE Mark II timesharing service, I think it ran on a GE 235. One of the early, early timeshare machines, and I kind of looked at it and saw it was BASIC and there was a manual there and I started to fool around with it and as soon as I'd typed in a line of code it immediately told me if I'd done it correctly or not and then when I hit run, it actually responded and I was -- it was like the heavens opened. Because remember the experience I'd had was this agonizing thing and it wasn't actually until UT brought in a [CDC] 6600 that they had any terminals attached to any computers at all there. You may remember they wrote Kronos operating system for CDC which is the first kind of big multitasking OS that-- and CDC shipped that to a lot of people actually.

But in any case, I was in love and I started writing programs and then I got the administration to let me teach an afterschool class in it. Then I wrote a little attendance program. In those days attendance was taken on punch cards and processed in the unit record equipment that you got sitting back there. Five fourteen reproducing gang punch [IBM 514 Reproducing Punch] and and old oh eighty three sorter [IBM 083 Card Sorter] and so forth and so on. And so what I did is I built a little program with all of my students names in it and I would-- I would call roll and I had a kid sitting next to the terminal in my class and if somebody was absent they would punch in that student number and then at the end of that process it would print out an attendance roll, which I would then hand to the student that came by to pick it up to take to the office and the office was like "What the heck is this? <laughs> How is he doing this?" So that plus the courses I-- the informal courses I taught led to a summer workshop first at University of Colorado, Boulder. That was an NSF sponsored workshop for teachers who were trying to introduce computers into the classroom. Now this would be '71, '72 kind of time frame and they had actually HP gear there that was associated with-- I'd say Cherry Hill or something like that school district there. So we spent a summer there, hacking around writing programs and starting to think about how we'd teach kids about this stuff. Then the next summer after I taught another year at Oak Cliff I actually worked with a professor

  

at SMU [Southern Methodist University] and we sponsored our own NSF summer workshop and that led to one thing and other.

But point is that the administration kind of saw what I was up to and at the time I was on an emergency teaching certificate and they told me I was gonna have to take practice teaching before I could-- now I'd taught three years at that point but-- and if I was gonna stay in the classroom I had to take practice teaching, I had to take a semester off and I was married, had a kid, no way I could do that. But the administration, Dallas school district administration, talked to me and said, "Hey we have this position where it's a computer liaison to the schools and you seem to know something about computers and stuff like that would you be interested?" So I moved to that role, there was one fun thing about that. I gave my students a goodbye present.

South Oak Cliff High School at that point was a state power in football and of course Texas football, particularly at the high school level is everything right? So I had worked with one of the coaches who was a part-time coach at the Dallas Cowboys on a computer scouting program for the high school team. A tendency analysis, pretty simple. The nice thing about high school football is you got a big tackle, you got a big fullback, pretty much. There are very few interesting flavors of things you do particularly in a third and short or something like that. So I had the coaches fill out a little form, which we punched onto cards recording every play. So when they went and scouted another team, they would record all the plays that that team ran and what the down and yardage was and so forth. I wrote a big Fortran program that basically crunched all that and then produced a tendency analysis for an opposing team. So I remember it was fun I was sitting-- I'd sit in the stands, "I think they're gonna go off tackle on this one," and, or they'd pass or something like that or draw play and coach had a copy and the kids-- I'm not sure it did much good but the kids were convinced it was mojo right? It was the secret sauce of its' day and of course in those days Cowboys had been-- basically pioneered that in the early '70s. Not just scouting of the teams but also scouting players and draftees and so forth. So there was that, so I had written that program.

So my goodbye present to my students was an interactive football game. So what you did is you punched in the play you wanted to run and then somebody else punched in the defense they wanted to call, without knowing what the play was, and then a series of randomizations occurred that were basically based on that combination and then it would record how far you got and what down it as and you'd start the process over again. Basically you could play pretty much all day, back and forth, and it was actually fairly realistic. The kids loved it and a number of those kids went on-- I remember one of my students, you understand this is an all-black school, and at the time talented minority kids were being recruited by all the top colleges. So I had a student who went to Dartmouth from Oak Cliff and I remember she sent me a postcard back of the Kiewit Computational Center where BASIC was born, Kemeny and Kurtz, and said, "Hey, just wanted you to see where I'm at after that." So that was kind of cool. Very rewarding stuff.

**Brock:** Oh sure.

**Pohlman:** But anyway I got into administrative computing, a couple years of that, and I got recruited by the local regional processing center, computer processing center. They had a giant, giant in its day, IBM 370-158. But they also had a little HP minicomputer timesharing system, HP 2000, and that's what they

  

wanted me to do. Let's take your educational computing, let's go back into the schools, say we're trying to sell time on this machine to teach computer math.

I did that, I put applications on it like the Guidance Information System. So we put terminals in guidance offices to talk careers and choose colleges based on early parameters. Of course this is all before, way before the internet so you basically had databases and that was about it. We did that, computer math, I started computer literacy, we started some programming classes for some elementary schools, fourth, fifth, sixth grade kids learning to program, of course high school. We wrote some applications, finally renamed the whole thing, the Instructional Computer Network, ICN. It was one of-- in those days there was TIES in Minneapolis, Region Four in Houston, there was Philadelphia schools and this Dallas organization that I had.

**Brock:** Now who ran it, who ran the whole operation?

**Pohlman:** I did. In Texas there were 20 educational regions and some of them were considered super centers where they basically sold computer services to the public schools. So up to that point you had a lot of school districts, particularly larger ones, who had their own computers systems. But it didn't really scale very well, and they were darn expensive and obviously very difficult to program.

So Texas established some regional centers and Dallas was one that basically school districts shared via-- if you remember the term RJE, remote job entry. So a school district would have a terminal into this monster IBM machine that did their grade reporting and their attendance accounting and all the rest of the stuff that-- finance, tax accounting and all that stuff that school districts had to do. We were fairly unique because we were also sharing student timesharing services in addition to administrative services.

**Brock:** So it's a private company, not the state?

**Pohlman:** No it was actually an entity of the state but it was-- it was not a government entity per se it was kind of funded by the state and blessed by the state but, so quasi-governmental.

**Brock:** I get it.

**Pohlman:** It still exists and other states kind of have similar kind of set ups. So we built that up, started putting HP machines in various remote school regions and eventually had hundreds of kids on hundreds of terminals I should say. So it was kind of an exciting time. A lot of those kids went on-- I had a student Fred, what was his last name, he went on to Southwest Technical Products and wrote their BASIC. He had written a Fortran compiler that emitted BASIC because the machines we had could only do BASIC so if you wanted to do Fortran you had to first translate it into BASIC. That was fascinating you had a COBOL interpreter, a bunch of other stuff like that. Another kid who-- another kid who wound up at Apple in their, in the original-- not the original Lisa group but the group that was trying to do with IBM the object oriented OS, Pink. So he came out of Dallas, out of that environment. So lot of fun, lot of sharp kids and at the time when it was a really pioneering era.

**Brock:** It seems like it.

  

**Pohlman:** And in fact, I think you and I talked about this, at a time when you could actually understand how computers worked it was like-- like today, for example, if you want to work on a car? Well you can maybe change the oil, maybe, right? But the notion of take a cylinder head off and doing any work on a car, long gone. In those days computers were like the four cylinder visible V8 right or the uh the eight cylinder visible V8-- you could actually see how they worked and you could write OS's for them and you could write compilers and assemblers and stuff. It was-- you got a sense of how it worked at the gut level, so it was fun.

**Brock:** It seems like things were going very well in that area, this educational computing area, for you. What prompted a change for you?

**Pohlman:** I actually got recruited. I had pretty high visibility within HP because I had bought a bunch of their stuff and particularly HP came out with a 2000/Access and I'm not sure but I may have bought most of them but it was ultimately not a hugely successful product. But it was certainly perfect for what we were trying to do and I got the-- I guess the attention of my local sales office there in Dallas knew me pretty well and I got a call because HP was looking. That was the old 3000 Division or GSD, they did a 2000 and 3000 series, and I got a call saying, "Hey there's an opening in education marketing in the division, would you be interested in talking to us?" And at that time the head of education marketing, I guess, had come to Dallas to see what we were up to and so forth and so on.

So I flew off, that would've been '77, yeah. I flew off and first time in California and it was kind of fun, it was nice, they treated me well. I got to work with, briefly interview and even be in a lab a little bit with a couple of engineers who asked some questions and I was able to answer them. The nice thing about that was if you wanted to program well in that environment you had to know the OS as well as the-- as well as the language. Because the timesharing system's pretty primitive in those days, time slice things. So if you knew how to grab more processing power than the next guy your program would run faster. So there were tricks you could do to fool the OS into believing that your application was really a low priority when in fact you could make it higher priority. Anyway I had done some of those tricks and written some programs so they were kind of curious about that. In fact they were trying to figure out how to defeat those tricks and so we had a good conversation and they offered me a job and I, I went for it.

Moved to California, I remember a long drive out and '77 was the height of the famous California drought. "If it's yellow, it's mellow" that whole thing, and coming from Texas, which most people think of as dry and kind of desert-y was not at all. It was green compared to this brown wasteland that I had driven into and I thought to myself, what have I done? But I had a good time, McCracken was running the division those days, Ed McCracken who went on to found Silicon Graphics, and I worked for a woman who was also a UT graduate, Carolyn Morris, who is one of the founders of the Silicon Valley Women Engineers Society. Fascinating person, very bright, had come up the line as an HP CE and worked her way into a marketing role, business development stuff. They were-- John Couch was there who was eventually Apple's VP software, Tom Whitney, who became Apple's senior VP of engineering was in that division. The place was loaded with talent.

So I spent a couple of years there, first in education marketing helping the company shut down the education market. They had decided to hand it to DEC and focus on manufacturing. I was not a fan of

  

that but I did it anyway. I wound up being the one who got all the letters to Bill and Dave that said "What are you doing?" In those days HP "owned" [dominated] a group called the HP business school user group. So this was [University of ]Chicago, the University of Virginia, all the famous schools. They did the Huntington simulations, I mean it was a very prestigious group, and HP handed it away. So I did that and then I moved onto data comm. and sales development; I knew a lot about data comm. So I spent two years there doing what we called the "quantum theory of management". You ever heard that theory but-- I'm not sure they call it there but a bunch of us wags called it that-- but their idea which was a wonderful management development thing was-- So you'd have a job in a particular division or group in some function and if you did well you'd get recruited to a similar level job in another function and then another function and then some "quanta" of management energy, blessing or whatever, would enter and you'd pop to a next up management role and you'd do the same cycle. Now if you'd done the loop a couple of times you knew that thing were not going well for your career but basically people popped up.

What happened with that at HP and it was instructive to me, I thought it was brilliant, was by the time you got to senior manager at HP you'd pretty much done every job in the company. So there were guys-- you'd cycle through manufacturing, you were in marketing for a while, you were in support, maybe in engineering, I mean whatever roles. So I went around a couple of, I went around once and actually ready to-- the quanta beam was coming down and then another beam struck and that was a friend of mine who had worked there who was-- actually had come, he'd done his rotation in GSD from the calculator division up in Corvallis. He had gone to Apple and he called me up and said, "Hey Apple's looking for an education marketing guy, are you interested?"

Well at that point I had been out of education but I still had a reputation that I knew how to talk to teachers. So I went over and took a look around and was pretty excited. In fact I remember going back to HP that day, after the interview and I still had my Apple guest sticker on <laughs> and my, and the admin in our department said, "I think you probably want to take that off before you see Roy," and I said, "Oh, oops." <inaudible> In those days it was so bad - I mean in terms of Apple's recruiting out of that division 'cause by the end Whitney had left, Couch was either left or was getting ready to leave. Whitney had put up a huge kind or hierarchical org chart on the wall of the engineering department with everybody's picture on it, right, and so many people left they started putting Apple stickers over people's pictures when they were recruited away. So anyway, yeah so I had my exit interview with, I think with Ed probably and I think I said this, he named my offer to within a few shares I think, they were so familiar [with what Apple was offering]. I was very nervous about it because I remember talking to my mother, who was working for the government at the time and she knew HP and she was back in D.C. and I was in California and so, but at least HP was this legitimate company that she'd heard of and I told her about Apple and she was like, "I don't know, that sounds pretty risky," and I started to think about that and maybe she's right and then Ed said, "By the way, if you do decide that it isn't working out, you've been doing well here, you can come back and you'll still have your seniority," and I went, "Well that's good enough for me, all right, see you later," and I just headed out the door. Actually they were gonna walk me out but I was in the middle of a project and so I couldn't tell anybody for two weeks while I finished the project and then got out the door. Went over to Apple and started out right away.

**Brock:** This was the computer division of Hewlett Packard that you were a part of?

  

**Pohlman:** Yeah there were two, there was general systems division which did the 3000 and the 2000, basically multitask timeshare machines and, in fact the 3000 was sort of a conceptual copy of the old Burroughs 5500 stack machine. I don't know if-- I didn't see a Burroughs out here but 5500 was the first virtual, timesharing machine pretty much ever built. That same stack architecture is what HP adopted for the 3000. But there was another division of HP called DSD, Data Systems Division. They did the real time computers, the little minis and 'cause remember HP-- When they shipped the 3000 in this giant box, it still said "Measuring Instrument" on the side right?

**Brock:** Right.

**Pohlman:** It took them quite a while to put the word "computer" in front of anything they did. But DSD really was, because of the real time, they had the Fast Fourier analyzers and that kind of software used a lot in process automation and so forth. But GSD was general purpose computing so they actually had COBOL and nasty stuff like that. Yeah, so it's actually interesting because within-- was it DSD at that point? I forget. The HP terminals, their programmable terminal actually was being developed at the time, the, I want to say 2640 Series, I didn't see one out there. But there was actually a mini BASIC for it, it was a great machine. Pretty expensive as a terminal but a wonderful thing and HP completely resisted turning it into any kind of interactive personal computer, the tiny BASIC somebody hacked in engineering. But it was basically with that screen it looked like a Lisa, it just didn't have a floppy drive associated with it.

But see, I thought that was the future. I had known that from the timesharing days. I mean the reason I was so eager to leave the timesharing organization for education [the ICN] that I had done and go to HP in the first place was 'cause I thought timesharing was dead and the reason was the IMSAI, those machines, the first 8008 based machines were gonna kill timeshare. I mean my customers were paying three thousand dollars a year for basically a teletype and access to this timesharing machine and the databases, right? But for fifteen hundred bucks, or less maybe, you could buy your own machine, capitalize it once and it was free from there on out and I thought, "Man there's just no way the economics [of timesharing] are gonna work." They weren't personal computers in those days but microcomputers were gonna kill time sharing and so I was looking to get out of the business anyway and then of course I was at HP for a couple of years and they were doing nothing about, what seemed to me, a pretty obvious trend.

So Apple was like, yeah, oh, okay and you know, clean sheet of paper, no legacy, just do it and kill the timesharing guys and they were already in education which is where-- in fact they were selling to the old HP timesharing base. So I could go in, I went to TIES, for example, I visited Region 10 [where I had worked]. I visited my old Region 4 in Houston, all these places that I knew from the HP timesharing business. When I was at Apple in education marketing and basically just like, "Here's your answer," and we had imported most of the Huntington simulations to the Apple II and so it was a done deal. The only thing Apple II didn't have was decent databases so there was still a role for machines like that but not for computer instruction, not anymore.

**Brock:** Two quick questions if I could. That general computer division that you were a part of, where was that located physically, where was your office?

  

**Pohlman:** So the first-- let's see that would've been Santa Clara so it would've been-- where was the building? It wasn't on the Wolf Road campus that was where DSD was so I believe it was the Santa Clara campus, which at that point was shared by the optical division. They made the little sensors and stuff like that. In fact there's a picture of that building when GSD occupied-- when GSD was formed, there's a famous picture of McCracken's office, which was a cube in the dead center of this huge, I mean we're talking maybe 50,000 square feet or something like that, crazy, and he had like a single cube in the middle and there were no cubes around it and so they basically built it out. Remember in those days, the only people at HP were-- that had offices with doors were Bill and Dave. So everybody was in this and in fact it was interesting because it was so egalitarian in that sense that all the corridors to get to the cubes were on the outside of the building near the windows. Nobody got a window office either.

So we broke a few of those rules at Apple, in fact that was, speaking of cubes that was my first experience at Apple. When I got hired, it was the old Bandley 1 building, 10260 [Bandley Drive] and I went in and they had just moved manufacturing out to Bubb Road and so the space that used to be manufacturing was where they were gonna put these new marketing people they just hired. But there were no cubes, no desks, no nothing, I said, "Well where do I sit?" They said, well and they pointed to the wall there was a stack of those old Westinghouse panels and an Allen wrench and so I built my office. In fact I remember a few months later somebody came in and was complaining, "How come Taylor's office is so big and mine's small?" I said it's pretty simple I got here first and took the big panels. That was pretty wild.

**Brock:** Was your office at Hewlett Packard closer to where Apple was located than to sort of the main headquarters of HP?

**Pohlman:** Well yeah, main headquarters of HP and the labs were all up in Palo Alto so yeah Apple was what? A couple miles away just right up on, at Wolf Road or-- Let's see that was where 280 and Stevens Creek came in is where that plant was and so yeah so it's maybe a couple of miles up the road.

**Brock:** To go back to that time in Texas when you saw the rise of, we'll call them the microcomputers and you have this realization that time was short for timesharing I guess, how were you becoming aware of those developments? How were you following kind of what was happening with microcomputers both in Texas and when you came to the Bay Area working for Hewlett Packard?

**Pohlman:** One of my vendors that I was getting, I guess DECwriters from and renting the teletypes, was starting to experiment with-- was starting to actually look at selling some of those [microcomputer] machines. So this would've been '76, '77. I was also aware, I think, I wanna say at that point Southwest Technical Products, in San Antonio was starting to build some machines from scratch. There was obviously inside-- so I was reading some of the magazines and that kind of thing. At that point, also in San Antonio was Computer Terminal Corporation doing some of the early CRT's right? So I had a couple of those and so I was kind of familiar with the scene down there. Most people have forgotten there's quite a bit, ARCNET came out of there - I mean there was quite a bit of stuff going on in San Antonio in those days and just my proximity to Dallas. But I was also-- I was looking at microprocessors, I was looking at a lot of things. It just seemed obvious at the time.

  

**Brock:** Could you fill in the picture of that offer to come do the educational marketing job at Apple? Who approached you? What was it like, what did it feel like when you got there? Who were the people that you spoke with? Just flesh that out.

**Pohlman:** Let's see, Carolyn Morris like I said, was the lead, she was running the sales development organization at that point and as I said we shared the experience at UT, she was in math also but I think applied math and she was a striking person anyway. About maybe six-one, six-two, beautiful woman, but with a bad habit of slumping because every guy she worked with was shorter and she had a tendency to wear heels and then, which made it worse. In fact, she and McCracken dated for a while and it was, cause Ed is maybe my height or maybe even a little shorter and so it was kind of like. I remember telling Carolyn like, "Hey if you're gonna wear heels, just stand up, come on." But she's a great lady and she had a vision for the thing. There was another guy who was running the education part of that marketing group, Gary Stump, who wound up I think at, up in Oregon. At another tech company, I wanna say Tektronix or something like that. Stumpy was a fantastic guy and just a gregarious personality and he had taken over education basically to-- engineering had cranked out a couple of products designed for the administration part of the education market and so part of the job at that point was to launch those.

So I remember talking to a couple of the engineering section managers that were responsible for those products and the good news is, I mean, it was a scheduling package and a couple of other administrative packages and I had done education administration and scheduling so I was very familiar with the issues. So they-- in fact I think the person who, I talked to the person who wrote the program and it was a very-- scheduling school classes is a hard problem and at the time a lot of applications were just brute forced it. So you'd fill up a class and oops that didn't work and you'd drain it a little bit and try again and stuff. It was nasty. Took hours and hours of computer time, expensive computer time. This guy had written-- like most HP guys, there was a lot of intellectual power there. So they'd hire somebody who understood routing theory and all this kind of stuff. So he had written this elegant thing in systems programming language, ALGOL [actually SPL] I guess at the time. He waxed poetic about it and I actually knew enough about it to appreciate what he was doing. We had this kind of little love fest and stuff like that. I was charmed by the-- I thought the people were really bright and I thought-- and I was ready to get away into a structured company. I'd been kind of running my own thing and-- but I was nearly thirty years old at that point and I thought working for a company with some sort of structure and stuff would be important, and California was a good break for me too. But yeah, it was an impressive group and the other thing I remember about it is how organized they were. So they had a folder full of stuff that said where I was staying and where my car was and they had this and they had a structured-- and I had never seen things so organized before. I was used to pretty sloppy kind of things, so I was like, "Oh, this is, wow, this is pretty cool."

**Brock:** What did it look like at Apple when you went there? I imagine there was a contrast.

**Pohlman:** Yeah. Well to give you a feeling, Eddie somebody or other, who was the HR guy at the time, I met him about a week after I joined the company. I remember this guy roaring up on a motorcycle, took his helmet off and said, "Are you Taylor?" and I said, "Yeah." He said, "Oh, I'm," I think it was Eddie. "I'm the HR guy. I think there's some paperwork I need to get you to fill out." Yes, there was. So yeah, that, building my own desk. The thing at that point was I was pretty comfortable with that. Again, I'd been running my own thing in the timesharing network and I was used to kind of making my own way. I wound

  

up working for a guy, Greg, I can't remember his last name now, [Greg Smith] who was the Director of Education and remember in those days, Apple had the Education Foundation. So they actually had a nonprofit they had set up in parallel to Apple's marketing effort that promoted computers in education and so he managed both the Apple marketing piece and the Foundation.

Greg was a traditional guy out of CDC and he was probably the most straight laced and organization man guy in the whole company at that point. He was also about ten or fifteen years older than anybody else and was not very technically oriented. He knew marketing. He knew the process and stuff and he had some background in education, which is why they hired him. But the rest of us were kind of wild people in his view. The thing I also noticed about Apple, like with HP, was that tradition of wandering around in engineering. So HP had the "management by wandering around" theory from the top-down. I was a big believer in that and that same culture existed at Apple in terms of "Go talk to anybody and find out whatever you need to do and just get it done." And remember, I'd been a programmer. I'd written a lot of code. I understood operating systems. I certainly understood the technology of the II and so-- or Apple II. And so I was pretty comfortable around engineering and my first assignment was of course the Bell & Howell Apple, getting the black Apple, getting that out the door. That was just a matter of getting UL approval for the Apple because Bell & Howell required that before they'd ship it to schools. At that point, the Apple II was not UL approved and in fact, we had to pay I think, I want to say twenty-five bucks a unit to LA, to the fire department in LA, to be able to sell computers in Los Angeles because they weren't approved. Nobody paid much attention to it. "What the heck?" It was that kind of informal kind of company. And there were only a couple of hundred people in the whole company at that point anyway.

**Brock:** That's what I was wondering.

**Pohlman:** So I had to get UL approval and the only obstacle of that was a guy that maybe you've interviewed, hopefully you have, Rod Holt.

**Brock:** I haven't, but the name is familiar.

**Pohlman:** So Rod was out of Atari. He was an absolute genius and invented the switching power supply. Rod did not suffer fools, I guess is the way to put it, and in his view, the UL power engineers, who were the ones who looked at power supplies for UL approval, were Philistines who couldn't possibly understand this brilliant thing he'd done, and he was probably right about that. But the thing about government functionaries is that they don't care. <laughs> It's like, "Okay, we don't get it and you're not going to explain it to us and you think we're stupid and so we're just going to sit on this thing."

Well, that was the impasse at that point, so nothing was happening when I took it over, and Bell & Howell was calling every day. "When are you going to get approval? We're ready to ship" and so forth and so on and it's like, it wasn't happening. So I found a young engineer who was not quite so prideful, who was actually willing to document the thing sufficiently for UL to understand what was going on inside and understand also it wasn't going to blow up because remember at that point, you were talking high frequencies and high voltages and it was not the sort of thing where you just took 110 [volts] and made it into 12 [volts] through a transformer and a bunch of rectifiers. And so we got it approved, took about six months. Bell & Howell was happy and we shipped the things and so forth, but it was interesting.

  

**Brock:** And that agreement was already in the works then if they were clamoring for it?

**Pohlman:** They had shaken hands. Somebody had figured out how to spray paint the normal case black and they'd even come up with a-- or actually, maybe I had to come up with the logo. I can't remember, but in any case, yeah it was a done deal, but remember in those days, there was no documentation. There was no process involved. They knew how to manufacture it because they had people who'd done that and knew what to do. If they needed more screws, they'd call up somebody and order some more screws. There were no processes in place and so to actually create a product, a new version of the product, and get it to a company like Bell & Howell who actually demanded documentation, drawings for example, "Well, where are the engineering drawings?" "Well, we have a sketch, would that be okay?" It was the first time that Apple had to explain what it did to somebody else, somebody who had rigid conventions and processes, who'd done a lot of outsourcing. On the flipside of that, Bell & Howell was a company that was dead from the inside. I'll never forget touring their Chicago factory. It is a big empty space and their labs, their labs were like HP museums of vacuum tube volt meters. I mean, the Apple labs' latest, absolutely latest hardware, engineers got whatever they needed that they needed to build. The Bell & Howell labs, they were obviously not doing any development, so it was kind of sad in one sense. On the other hand, they knew how to do stuff, process stuff, so that was a discipline that Apple needed at the time, and it really helped.

**Brock:** And they also had existing contracts into the education world? That was the whole...

**Pohlman:** Well remember, yeah, they were selling projectors-- well, the thing they had is a salesforce, they called on education, so the projectors and the overhead things and all the rest of the stuff. Apple was looking for a distribution channel. Remember at that point, you just had a few hobby dealers. Ron Rohner, who's my partner today in the consulting business, was the one who built Apple's first channel. He came out of consumer electronics and he worked for-- I'll remember that, the VP of sales, Gene Carter. He worked for Gene Carter -- but Apple had no sales people in the field and the dealers they had were mostly hobby dealers and so there was nobody calling on education per se. So the notion was get a distribution channel, get somebody who's calling every day, put something in their bag that's this new technology and so forth. By the time the deal was really done though and by the time we got it out the door and by the time they got some traction, Apple was also building its own reputation in the education market and so it didn't take too long before the deal kind of became irrelevant. But it was again, good discipline.

**Brock:** How big of a target was the education market for Apple at this time and maybe its competitors to the extent that it had them versus the office?

**Pohlman:** Well, the office really didn't exist as a market at that point. It wasn't really until VisiCalc in late '79, early '80, that that became a feasible target. So education was important because what the Apple II could do, which was deliver BASIC programming experiences with color graphics, was exactly what education was looking for from the standpoint of computer literacy. And the big push at that point was computer literacy and it didn't require disk drives, it didn't require a whole bunch of stuff that business applications did. And the competitors at that point were reasonably successful in education. You've got the Commodore PET, you had the TRS-80.

  

The thing about those machines was the PET was significantly limited in a number of ways from a programming standpoint, memory and so forth, and I remember when Apple introduced Pascal, the language card and UCSD Pascal, it took a step also into higher education where those other products were not competitive. RadioShack-- remember it was, what was it, I want to say late ’79 or early ’80, {actually late 1980] yeah, when the FCC stepped in and required personal computers to meet [RF] emission standards and that was the death of the Trash-80, or the TRS-80, pardon me. Don’t speak ill of the dead. The TRS-80 could not be made to pass FCC because it had too many pieces and wires and boxes and stuff and the PET had serious problems too.

With the Apple II, I was the Apple II product marketing manager by that point. We figured out how to coat the inside of the Apple II with some metallic paint, ground it in a number of ways and stuff and we actually got the, almost exact unit to pass FCC. In fact, funny story about that, and also about Rod Holt. Rod was, as I said, a genius and part of it was because he thought in different ways than most people. So at that point, to test for FCC approval, most people were creating at enormous expense, these radio-wave dead rooms, shielded and so forth and so on, so they could test the emissions of the box without interference.

Well Rod said, the problem with building one of those rooms here in the Valley is there’s too many radio waves already, microwaves, everything else. So Rod bought an RV and went out to north of Santa Cruz, along the coast where everybody knows you can’t pick up a radio signal for life. I mean, you drive along and your AM radio is just useless. There are no signals because you’re completely blocked by the mountain range. So he just set the stuff up in the middle of a field. Somebody else is spending a million bucks, he was spending whatever it was for the RV plus these sensors, and he just set the Apple II up on a thing in the middle of a field, put his sensors around in a circle and he was done. And then of course, we took the result and we knew it would pass, and handed it to the FCC who put it in one of the rooms and actually verified that it worked. But I don’t know many people who know that story, but that was the way his mind thought. It’s like, “Well, why would I solve a problem that way when I can solve it this way?” So anyway, but that was some of the story there.

**Brock:** So it sounds like you were just in that educational marketing role for a brief time then, less than a year?

**Pohlman:** Maybe a year or so. Yeah, what I did was I came in, I did the Bell & Howell Apple, the FCC thing-- or the, and the-- not FCC thing at that point, the UL thing. I also got Pascal out the door. I was the product marketing manager for that, and I’m proud to say the marketing manager for the Apple Pascal poster. And as I think I told you earlier, I still have one that has Niklaus Wirth’s signature on it that John Couch got for me in Switzerland when he saw him. So I got Pascal out the door and that was a big breakthrough for higher ed. Ken [Kenneth Bowles] down at UCSD, he had done the p-Machine [UCSD p-System] and all that and that was a very expensive kind of choice. We built a language card, extra 16K, stuck it in the Apple II, and bingo, we were off and running. And as a part of that, we did 16-sector DOS, which was another whole story. But in any case, having done that and getting a little bathed in the hardware, Steve was running-- Steve Jobs was running product marketing at the time and Phil Roybal was the Apple II guy and Phil was moving more into a more general marketing and customer service kind of role, and so I got tapped for the Apple II job. And part of that was-- well even then, Apple II was

  

obsolete by 1980, from a technical standpoint, 6502 was-- although it was one of the first RISC processors, it was hardly leading edge at that point.

So the Apple II was really all about making it work in volume and I think I demonstrated that I could get stuff out, I could get stuff documented and I could work with engineering on that. We had the 16-sector DOS to do, which was important to software development at the time, because remember, programs were starting to show up that actually did business stuff. So you had Mike Markkula with the stock package. You had Apple-- the little Apple Plot package. You had a couple of-- Don Williams did not have Multiplan at that point, but the precursor to that, Desktop Plan. And then the Controller [accounting software]. So when Apple shipped the disk drive, the first floppy disk, at this insanely low price because of Woz's magic state machine.

When Apple did that, it built an era for software development on that platform and stuff started to appear, and then of course VisiCalc as well. Well, to do that, you needed a decent, reliable disk system. You need enough storage. So the Apple II was really all about taking the technology as it stood and just making it work reliably, constantly. We had early problems too. The disk drive had media change and so suddenly disks started to fail. A lot of it was just basic quality management and blocking and tackling, and the other thing that was going on, the channel was evolving so you had resellers and dealers who were actually starting to look at things like accounting on it as a platform, and it was starting to steal share from the bigger-- the more classic business machines. It was cheaper, more reliable, more profitable for it and so forth. So it was really about execution and that was fun for me.

The other thing it was about was I think the characteristic of the Apple II was always kind of a lack of hubris. You think about the characteristics people think of Apple, particularly which started with the Mac and actually before that, even with the Lisa. It was like, "We know what you want to do with a computer, pay attention and we'll explain it to you." With the Apple II, it was like-- and to a certain extent, the Apple III, although the early issues-- so the Apple III marketing also had to do with hubris. We can talk about that in a minute. But particularly with the Apple II, it was like, "Here's this cool thing, here's how it works, why don't you see what you can do?"

So for me, the most exciting-- for example, the classic was the Adam Contest. You may have heard about that before. "What in the world would you do with an Apple II?" So the idea was a contest where users tell us their most innovative application and we have a prize for the coolest app. The guy who won it had taken an Apple II with an analog board, A to D board which you hooked to a mic and an analog board out, motor control boards hooked to some servos. You put the mic in the bedroom where his colicky baby was and he had written a program that basically said-- figured out some crying patterns or noise patterns in the night and if it met a certain criteria, the servos would go and rock the cradle.

**Brock:** Oh my gosh.

**Pohlman:** Well, as a marketing guy, you don't want to say, well, talk to a bunch of engineers and say, "Hey, I think we could build this thing that quiets babies." So nobody was going to think about that, but we gave people the tools, the analog boards, the motor control boards, the mic input, the programming language, the so forth and so on. We gave people all the tools so that their creativity could be featured

  

and so it was that lack of hubris. I had a line in the Apple IV business plan about building a new technology platform like that so that "users can continue to surprise us with what computers are good for."

In fact, I used to call-- before it-- well before it shipped, when it was first conceived, when Steve first took the idea from Jef [Raskin] for the Mac, I called it "TCTDFT," The Computer That Does Five Things. And for me, it was a joke because why in the world would you want a computer that just did five things? By not putting a language in it, by not giving it enough memory and so forth, you basically made people use it a certain way when people actually wanted to do a whole bunch of cool stuff with it. So that kind of tension within Apple, the hubris of "We know what to do" and the faction that said-- remember, when Woz put the Apple II together, and for that matter the Apple I, but before that, particularly the Apple II, it was the Redbook, all the Peeks and Pokes. "Here's an interpreter, here's all the stuff, go for it" and they did.

**Brock:** Thinking about those two camps of kind of the hubris and the one right way and then the kind of platform, open platform, and you have two Steves to choose from and you could easily see-- Were those the two camps? Did they kind of gel around that?

**Pohlman:** I felt like it. And it was one of the reasons why I left in late '82 and one of the reasons why I did software instead of hardware. I had a couple of offers to go create the next big clone or whatever and I had spent my career at Apple, such as it was, three years or so there, trying to make sure that nobody except maybe IBM could get into the computer business, the personal computer business. So software looked like the only thing, but the thing was that yeah, there wasn't really a place to go to build another Apple II. What the Apple IV was, of course, was a follow onto the III, was a [Motorola] 68000-based, open color graphics. It was basically a Sun.

I realized it a couple of years later when I looked back at the old Apple IV business plan is that we were really trying to build a Sun box, because it had a Unix in it. We had a couple of sources for non-AT&T Unix. In those days, there were a couple of basically work-alike Unix versions, so I thought Apple could probably ship a Unix that wasn't going to interfere with AT&T's rights and put it in an Apple II style open machine, 68000, and we were going to aim it at scientists, engineers, statisticians. Remember, Apple had built, I think I mentioned this to you, the SANE environment, the Standard Apple Numerics Environment, the K-C-S [Kahan, Coonen, Stone floating point] mathematics. So I knew that engineers, analysts, stock and bond guys needed a platform to do their creativity with and then software developers needed that kind of platform. So we had this notion but it didn't work out.

**Brock:** To take a step back a little bit...

**Pohlman:** <laughs> Am I wandering around?

**Brock:** Well no, not at all. Maybe my mind is wandering. But no, just to go back to something, it seems that when you got that job with the Apple II, that's quite an important job for the company. My recollection from what I've read, that's where the money that the company had was coming from, correct?

**Pohlman:** Yes, but it was also the place that-- it was the mundane place. The thing was you have to remember that the priority at that point was all about the Lisa and the Mac. First, the Lisa and as it

  

crumbled, the Mac. And so the Apple II was simply a place-- and the Apple III for that matter, because we got the revenues up pretty strongly by the end. That was my last job before I left. But the notion was that was just the bank that people drew on.

So if you looked-- in fact, I remember a funny story. Well, two stories. Apple employees always dressed up at Halloween. That was a famous tradition and people wore costumes and so forth and so on and pranced around campus and so forth. At that point, we were all over in-- so this would have been eighty-- probably Halloween ’81, ’82. I think we were in Bandley 3 at that point, almost one hundred and eighty people packed into that one building, five by seven cubes. It was pretty grim. Across the street was the Lisa building, I want to say Bandley 4, and so we were all dressed in costume, frolicking around and so forth and so on. The Lisa building, at that point, you had to have a special badge and a sticker on your badge to get in. So none of us had ever been in the building. At that point, I was the product marketing manager for the II and the III. And so I led this band. I think I was one of the Andrew Sisters that year. And so I led this band across the street and we simply walked, danced really, right past the guards and into Bandley 4, the sanctum sanctorum of the Lisa group. That was in the Trip Hawkins days, who went on to Electronic Arts. So all the folks that worked for him were similarly, they looked like a basketball team. They were all kind of tall and lean and scholarly looking and wore a coat and tie and stuff like this. So we dance into this building and what we find is indirect lighting, a fountain, ten by ten cubicles, kind of quiet, restrained atmosphere and so forth. And we looked around and again, we were coming from this nightmare of five by seven cubes. We looked around and we said, “Oh, there’s two Apples and we’re in the other one.” So literally, quietly, filed back to our offices. I was shocked. When I walked into Apple, again, having built my own cube and wandering of engineers and dropping into Markkula’s office to talk about stuff and hearing Whitney and Scotty argue in the hallway, it had become a completely different company in three years. I was shocked, and so were the people that were with me, and that was the other thing that probably left me-- it led me to exit at that point because the company was being run by people who, in my opinion, didn’t know enough about computer technology to know the right thing to do. That was, I think I told you, in ’81, I met with Bob Metcalfe and Bill [Krause], his CEO, about putting an Ethernet card in the Apple III, which we really needed-- we had two kinds of networking at that point, Omninet and the ARCNET stuff, and badly needed Ethernet. And of course, the Apple III was the perfect operating system for Ethernet. We had device drivers with multiple entry points. We could do networking. We could do a network printer. I mean, it was all built into the OS. All they had to do was write a driver. And Metcalfe was impressed, but then they went over and saw the networking experts in the Lisa group and got talked into the serial interface box, the thousand dollar Ethernet interface that took ten megabit Ethernet in one side and spit out one megabit serial on the other. It just was nonsense and so it was just-- yeah, my opinion was people who were running the company didn’t know what-- didn’t know enough about technology to understand the right decisions. And that was the-- it’s interesting, sales is always the lagging indicator. So what you sell today is what you invented a year or so before and if you’re not inventing anything today, you don’t quit selling today. Sales happens two years in the future [from invention]. So what was happening in ’83 led to the disaster in ’85, pure and simple, about a two year lag. The success in ’83 was all about what had been done in ’81.

  

**Brock:** Well, let's talk for a little bit about the Apple III project and how you came to it, your role in it. It seems to me that if this is happening-- well, was the Apple III project happening simultaneously with the Lisa project? Or it preceded it?

**Pohlman:** It'd come a little before. Actually, when I came in '79, it was pretty well set.

**Brock:** The III?

**Pohlman:** Yeah and Steve-- most people don't realize this because of the revisionist sense of history that surrounds Apple, but Steve [Jobs] was deeply involved in the Apple III project, down to figuring out how many bits the A to D converter was going to have. In fact, I remember wandering in his office one time when engineers were playing various tapes of voices at four bits, six bits, eight bits and so forth. And the case had been designed. Wendell [Sander] had figured out all of the stuff and the OS was just about built. Tom Root had just finished about that and device drivers were being written and so forth, '79/'80 time. Remember, the product launched in the spring of '80 at NCC.

**Brock:** Okay.

**Pohlman:** I looked at it as the Apple II guy at the time. I mean, it was my "competition." So a good marketing guy wants to know-- I mean, there was nothing else in the marketplace that was particularly competitive. It was a significant engineering leap at that point and so I paid a lot of attention to it. Unfortunately, the marketing team at the time, who were some ex-chip guys, didn't understand systems, didn't understand technology, didn't want to talk to engineers because they kept promising stuff that the engineers couldn't deliver and so engineers were mad at them, so they didn't want to go back and talk to those guys because they'd get yelled at. So I could see kind of the disaster looming, but-- because I did talk to engineers all the time so they would, <makes noise.> <laughs> But understand that the anticipation and the rumors were high. I remember going to an Apple user group meeting, so the heads of like Apple Pie and all those guys were in a special screening of the Apple III prior to launch at NCC, and they were all over it. They wanted to know how it worked, they wanted to know this, "What about this OS?" "What's this all about?" and so forth. And there was no technical documentation. The marketing group had decided that what they were going to do is a word processor, "Word Painter" they called it, and something else, a numbers package or something like that. I remember, one of my guys dubbed it the Painter Sisters, but I can't remember what the other sister was. So they were going to put some business applications on it. They had a port of VisiCalc that Bricklin and those guys had done, and in fact, it was going to be a super VisiCalc because you could have large VisiCalc models bigger than anything else that you could put on a PC, larger than 64K. So they were going to make it a business machine and therefore from a hobbyist standpoint, they felt that picturing it as a technology based machine would be harmful to the business positioning. It's called Business Basic because they didn't want people to think they could do anything else with it. And so as a result, there was no OS documentation. There were no device driver manuals. There was certainly nothing equivalent to the Redbook and in fact, that part of Apple marketing sort of aggressively tried to suppress knowledge about how it worked. They said Apple II cards wouldn't work when they would. They said a whole bunch of other stuff. And so net-net is that it launched as a much less interesting machine than it actually was. The OS was completely innovative. I mean, it was basically single user Unix. You could look at it that way. It had a higharchial file system, it

  

had not only device drivers, but it also had event fences, both software fence and a hardware event interrupt system. It had named memory segmentation. It was head and shoulders above anything else and it was called SOS, as you know, because of Sara, Wendell's daughter. But the marketing guys, genius, said, "What starts with S? Sophisticated." It was like, "Oh good grief." I remember just being-- structured maybe but not sophisticated. But anyway, so basically they fizzled the launch because they failed to think about how much energy there was in the Apple II base to have a better, more interesting, more powerful machine.

**Brock:** And they felt that that would be contradictory to this business market where it was supposed to be just sort of a plug and play, it just does these things?

**Pohlman:** Yeah. At that point, they were already drinking the Lisa Kool-Aid about a closed business machine. So the problem is that it's hard. It was still a 6502 and it was still-- I mean, it had a sophisticated memory management capability and so forth, but they called it bank switching instead of its correct addressing, so people dismissed the memory management of it. It was a failed launch by people who didn't understand the technology. And then of course on top of that, the darn thing didn't work from a hardware standpoint because they'd tried to pack too much stuff into it and of course Steve declared that it couldn't have a fan and therefore, it ran too hot and etc. etc. So by the time I took it in late '81/'82 to do the turnaround, what we called "Let me reintroduce myself," and got the sales back up, it was more-- it had gotten a stunted launch and the IBM PC got in. It was much less interesting technically, the IBM PC, than the Apple III, but by then, the company wasn't telling that story and they'd missed a year worth of lead in the marketplace.

**Brock:** Did you clamber to get to do that relaunch or did somebody come and grab you and say, "Please do this"?

**Pohlman:** I don't know if it was Black Wednesday, Black Friday, whatever that thing was where a bunch of people got the axe at Apple in '80. At that point, I inherited the customer service operation for the III and we put the turnaround program together. I got about a three million dollar budget and I had gotten-- the II was in pretty good shape. We'd started the IIe project and I'd given that to Mike Connor, and so I moved over to the III to do the turnaround. I can't remember if I was asked to do it or if I was just the last person standing 'cause it was not a popular thing. One of the marketing guys had left, the head of Apple III marketing, Don Bryson had left and the other guy, Barry Yarkoni, had moved over to the disk division at that point to do ProFile and so there was nobody left, so I started that process, we got some training out, we got documentation folks started on the device driver manual and the SOS manual and so forth, and I hired a guy named Jerry Bower who did the actual customer service part of the turnaround. He was, I remember his resume fondly, he was from Cornell, he had been in the computer business for a while but his degree was a Ph.D. in taxonomic entymology. Basically how many hairs are on the back of a fly determines which species it is or, you know, what genus it is, whatever. And I thought, now there's a detail guy. And he was, he was an animal for detail. We got a stack of Apple IIIs that worked and people wrote in and they said my Apple III doesn't work so we'd ship them a new one and say please send the old one back, and if that one didn't work we'd ship another one, we had some people that got four or five of them. And by the end of that process, people started to realize, holy moley, these guys are willing to spend anything it takes to make me happy. And some of our biggest detractors became our biggest

  

allies. And remember, I mean the III fully loaded with disks and a printer and everything else, it could go out at ten grand. In fact Portia Isaacson said in '81 I think it was, it had 50 percent share of personal computers over 10,000 dollars out the door. And, so the people who bought this were, you know, not only well-to-do, you know, had important things to do and righteously angry about their investment. So we got them turned around, we got some software, created the first development support group under, actually if you haven't interviewed him, Bob Martin, he wound up as Apple's legal guru, he went back to, got a law degree I think and wound up in the legal department working on things like the Franklin suit. But Bob was the head of the first software development, third party software development group and licensing group working for me in those days, and hired some Apple stalwarts, people like Russ Aldridge and Brian Stearns, I think Brian may even still be around Apple, hired him as a 17 year old kid out of high school. And, so we did that, we did some evangelism, getting things like Great Plains for example on the Apple III, and briefly turned it around but then the XT was really the death knell.

The XT and of course DOS 2.0 from Microsoft, which borrowed most of the important features of the Apple III OS, so DOS 2.0 for example had a heirarchical file system strangely similar to Apple III's OS, and it had device drivers strangely similar. It might have, well, I won't start any rumors but it turns out that the guy who had Microsoft-- Microsoft had the source code to Apple III OS, I don't know if most people know that but they got that for-- to do the CP/M board project so, in '82 they had the source to Apple III OS. But in any case, DOS 2.0 and the XT with the hard drive in it were pretty much the death knell for the III. There was a III Plus which I had kind of gotten started before I left, but engineering really wasn't funded to do anything interesting with it. But it still went on and sold for a couple of years.

**Brock:** I wanted to ask you, while you were doing these efforts and there's the Lisa - Macintosh sort of phenomenon happening, what you made of that origin of the Lisa and then the Macintosh products, what was happening at [Xerox] PARC, the graphical user interface, object-orientated programming, just that whole development around the Alto and what you thought of that personally.

**Pohlman:** Yeah so I looked at the, well the Lisa was, the hardware was pretty well set by '81, in fact I don't think hardware design changed from '81 till its launch in '83, which is unfortunate because they should have upgraded the processor and some other stuff, it was way too slow. The Lisa group, the OS there those were smart guys, they were real computer scientists, and the original Lisa OS, which was intended to be multitasking and some other stuff, their ambition got shot down and it also got kind of lobotomized from a memory management standpoint, but the people who did it were smart and I had a lot of respect for them. The applications they wrote were, I mean it was difficult because they were trying to sort of launch it on the QT which meant they had to do internal applications, they couldn't turn it loose and let, you know, people develop on it and it really had a very primitive development environment anyway. Sort of bootstrapped to yourself with it. But I was very impressed and, but I could see, because I was also a hardware guy and I could see that they were hamstringing themselves because the hardware was already obsolete years before they actually shipped it. Well I mean look, the Mac came out with a 8 megahertz processor and the Lisa was still, you know, at five I think it was. Plus they didn't do, they didn't really think about hard drives, they should have built one in from the beginning and they did this, and of course the whole Twiggy thing was, remember I told you Rod Holt was a genius? So Rod Holt was also the guy who told them that the Twiggy, which was their, that was the 800 KB floppy that had

  

opposing holes in the disks right, it had two heads right? What Rod explained to them is they were going to have to use laser interferometry to align this thing, and sure enough it turned out to be darn near that hard of a problem. They were not paying attention to the technology. They had a chance to get, in fact, if Apple had decided to invest. Actually we'll come back to that story. Apple had a chance to invest in Seagate and buy drives forever at a very low rate, they had a chance to get the manufacturing rights from Sony on the three and a half inch floppy which they passed on 'cause they were convinced, in fact I remember Steve saying, "we're going to blow them out of the water when they see Twiggy". I said, but Steve it's kind of cool, it fits right in your pocket. But anyway, a lot of technology decisions there made by non-technologists that kind of hamstrung the company in that era, but Lisa was impressive, this is your real question, and I could see how Mac was going to borrow some of that. And in fact we had mocked up some things on the Apple III, Wendell actually had designed a 68000 board for the Apple III using virtual memory notions and I had worked with SCO [Santa Cruz Operation] to, in '83, to look at a port of Unix for that card, for the Apple III. Larry and Doug [Michaels] there were fascinated by that idea. So there were, and Herzfeld was supposed to do a port of the Mac ROM stuff to that board for the III but then got told not to do that, which was probably smart because he was way behind on his regular job.

But the key was, so that was a technology that was interesting, and when I left Apple in late '82, for various reasons including the fact that I really thought there were some more interesting things to do, in founding Forethought, Rob and I, Rob Campbell who had been the Apple II and Apple III software marketing manager, and had done the Controller and some other software like that, both of us were convinced that that interface, and that technology and that approach to applications was the way the world and the future was going to be. And so, what we saw was it was just, the company was taking a great concept and putting it on junk hardware. Well, I mean, the Mac. So, all right, so "real men program in assembler", right, that was Herzfeld's approach, and why would you give them more memory so they can write compilers and stuff when, and there was actually a bit of genius about that with the Mac and that was, there was so little memory that you had to use the ROM to do anything, right? So that part was good in a sense, there were a couple of people who clobbered together non-Mac kind of style programs on the hardware but it was very, very hard to do anything useful, so you were kind of pushed into the ROM which was probably not a terrible idea.

The Lisa didn't sell enough to create a platform and that actually hurt Forethought when we started out because we were hoping that would be a platform for our software. But in reality they hamstrung the product, they cut off all the important leads on the SCC chip on the Mac so they couldn't do HDLC, they couldn't emulate [IBM terminals], they had 512 bits across, which made no sense for an 80 column emulator, it created hell doing 80 column terminal emulation on the Mac because you had to do seven bit characters to get 80 across, and it was just crazy stuff like that that, from my perspective, didn't make any sense, I mean why would you deliberately limit yourself in building a hardware platform when you are trying to do this insane software?

**Brock:** But the software approach of the graphical user interface and all that, that you, when you saw it, made sense?

**Pohlman:** Yeah, absolutely. Yeah, I mean it was, it made sense because, and interestingly enough, one of the most important illustrations of that was something that was kind of almost, you know, by the way

  

from a software technology standpoint, and that was called the grayed-out menu. So, if you think about it, and remember the two rivals for object-orientated or graphical user interfaces in those days were VisiCorp's product VisiOn and the Mac-Lisa interface, and of course it was Microsoft in there with their red herring of Windows, you know, the stub version. But the notion there was, and there was also remember the big philosophical battle in those days, object-verb, vs. verb-object right? So, do I select something first and then decide what to do with it or do I decide what to do and then click on what I want to do that to? Even at PARC there were opposing camps about that, the guy VisiCorp hired was in the verb-object group and the Mac and Lisa were all about subject-verb. So, point is that, if I selected something, then only the things that I could do to that wound up being available to me from a menu standpoint.

Now you've got to understand up to that point I'm sitting at a command line, you know, interface, I can type anything, so the number of mistakes I can make are nearly infinite, and therefore the set of error messages and prompts and stuff like that is also right, there's no way to steer somebody through a command line interface. Now, you can argue a command line interface for somebody who knows it really well is one of the most efficient tools ever invented, I mean I watched Peter Rowell edit in some of the Unix editors and just flying fingers, amazing stuff. But, for the average user, once I select a piece of a paragraph and I notice that all I can do is change the font, change the size, change this, as opposed to save, delete right? So, and I thought that was genius, because it just simplified the way you could attack software in a user interface. The rest of the stuff was really nice too, and the notion of working with what you see is what you get, I mean we'd struggle with that on the Apple II and the Apple III, all kinds of tricks to try and figure out how what you typed sort of formatted itself on the screen, it was a mess. But, and I had done, in my basic programming series on the Apple III I'd shown people how to do proportional fonts on the III and so forth and do a proportional font editor but it was a struggle. So that was the right thing.

And when we started Forethought it was deliberately to do that software, in fact I think I mentioned to you the whole notion was we knew that was the way to go, we knew that was how to build better software, and, but we also knew that the hardware platforms weren't quite available yet, we were hoping for Lisa to be successful, it wasn't, the Mac was too small and too limited to do the kind of software we wanted to do because our notion was something that's now called compound document oriented architecture, so a compound document is basically I have a bunch of document style objects, I got a table, I've got a bitmap, I've got some text, those are all different flavors of objects that assemble to form a document. The only way you could do that on the Mac, and for that matter on the Lisa, was "dead" cut and paste, so I could put a graphic into a Word doc let's say, but I couldn't edit it once it got there. Nor was there the kind of publically subscribed kind of link that said if I changed the original it would change there either. We thought both those were going to be critical.

And so our first business plan talked about object-oriented software but it talked about live objects, and to do that we had to invent some interface stuff that, at the time, wasn't available on the Mac and Lisa and is now available, and that was the notion of, I have to be able to poke an object and have it tell me what it'll do. So the same notion of the subject-verb, that could only be, live objects and active objects in a compound document could only work if you could do subject-verb because you had to say, hey, tell me what I can do to you so if I poke a table I get a different answer than if I poke a paragraph, or if I poke a bitmap, right? And so, the notion of a dashboard menu that pops up that says here are your options,

  

yeah, pull downs are fine, but at the end of the day, if you're selecting objects in a document you want a pop-up and, so we invented the notion of dashboards and a bunch of stuff like that.

Unfortunately the programming environment wasn't up to the task of building those kinds of things, the geniuses we had from PARC, Peter Bishop and Peter Rowell and guys like that were writing object-oriented code in standard C, there was no C++, remember I mean, the tools to build object-orientated environments in those days really didn't exist, and the stuff that the Lisa and Mac groups built was not object-orientated in the classic sense. So for example, it was a long time before there was a two-byte editor for text on the Mac or, to do Kanji for example. So, I mean, what [Randy] Wigginton wrote was a pretty straightforward word processor, just really nice looking in the output. So it was a formative time relative to the, you know what comes first, the application or the tools, right? Back in those days it was hard to find compliers that could, you know, address the amount of memory that you had, one of the things that the Lisa group did-- what Couch did in the Lisa group did is contract with a bunch of compiler writers who could build the kind of tools it took to do that kind of code.

**Brock:** Before we perhaps dig into really, you know, leaving Apple and founding Forethought and where these ideas, how these ideas for Foundation really gelled and came together…

**Pohlman:** You know that term, you must have talked to somebody else. Yeah Foundation was, that was the name of the platform.

**Brock:** Yeah, right, I can't remember who but I picked it up somewhere. I wanted to talk to you a little bit about, if I could, just about the Apple culture in those, you know, up until the time you left in late '82, and stories that you may have about working both of the founders, what those experiences were like, just about kind of Apple lore from those eventful three years that you were there.

**Pohlman:** Well remember, Apple culture formed out of the three cultures. So, there was the Intel culture that was confrontational and you were expected to be strong to survive, right, so people got-- there was the HP culture which was like nurturing and stuff like that, but appreciative in their roles, sensitive to careers, things like that, but also not particularly aggressive. So, HP's idea, for example, and some of this came to Apple and got killed quickly but, when I was at the old GSD division, HP's notion of engineering to develop a new product was that you searched all the best grad schools and found several students who were doing their Ph.D. thesis in that particular technology. You hired them, gave them a couple of years to kind of acclimate themselves and, you know, get, understanding what's going on, and then you ask them to produce the product. And what you got was the perfect product, completely, exactly great product, three years late. Other companies' approach was, and Tandem's a good example, is like hire a guy who's done it three times before and then sit down, get it done, out the door, right? And so, on the other hand it wasn't necessarily a great product, it wasn't very well thought-out, so at the end of the day HP would tie in the marketplace. So the culture of, so the HP culture came to Apple via people like Tom Whitney and Couch and so forth, the National culture came to Apple via people like Scotty, Mike Scott, right? Not only confrontational but, you know, very aggressive and so forth. Intel was more of a cool version of that, right, intelligent and so forth, but also chip-focused, so neither the National or the Intel people understood systems very well but the HP people really understood systems and architectures. So you talk to an HP person and they'd first start with a design and then they worked the

  

product within that kind of concept, and some of that came from the lab stuff right, HP invented a lot of the technology they ultimately used in the products. In fact we used to go on little tours of HP Labs in Palo Alto and you'd poke through and see, oh is there anything interesting this month to, that we can turn into a product, 'cause they weren't worried about product.

Those cultures hit at Apple and clashed and created its own kind of blended culture. So it became brash and it became get things done stuff and stuff, and on top of that, the founders, remember Woz had come out of HP and he was, oh I wouldn't call him part of that, he was a tinkerer more than an architect, but he understood innovation and he respected other engineers. Steve of course was not a technologist and--but he understood how to get things done and he survived quite well in the aggressive part of that culture because he, you know, he'd argue with anybody of whatever stripe and so forth so, so for me the part that was good was the, was the desire to do cool things, right? And that was still there. What came about, and kind of froze the culture in places, think about when you hire, right, you start with a few people and they hire their friends who they know and trust and know are cool people and smart people and stuff, and then you go outward, and I like to think of, in a small company the culture is that everybody is faced outward. Your competitor is, your real competitor in the marketplace right, is somebody who's trying to defeat you and your company, so you're circling the wagon in a sense but everybody's faced out and so, you don't worry about your back because nobody else is looking at your back, they're all looking at another issue or another problem right? But as the company grows, any company grows and it's certainly true for Apple, that circle expands and it gets filled in with people you hire through standard processes that nobody knows and doesn't even know if they're any good or not, and who do they face? Well they're not facing a customer anymore they're facing each other, so their competitors is a person in another department or in their same department, and suddenly the internal competition begins to eat away and the only, at the end the real Apple culture in, that I saw in '83, '85 that I really liked, was the field, 'cause they were the last people in the company who were actually facing customers all the time, and competitors. And so, the Apple field, and that's why typically sales forces have a different culture than the company anyway, but in Apple the keeper of the flame of the old Apple for me ultimately in the mid-80s was the sales force.

**Brock:** Interesting.

**Pohlman:** Because they still, there were a lot of, still people around, but they also were, like, genuinely just interested in beating competitors as opposed to each other.

**Brock:** Right. And what about working with Steve Jobs? There's a whole genre of storytelling about that, I wondered if, what was most meaningful for you.

**Pohlman:** Well, I mean he was a force of nature. I mean I, when I went to, actually I like to think almost that he hired me because when I first went to Apple as I said, I came from HP but I had known other people that went to Apple so, as young as I was, I was smart enough to know that I was supposed to get some stock, and so when they made the first offer I kind of assumed that was too low and so I asked for more, and that's what led for me to be interviewed by Steve because he was the one that had to kind of sign off on any more than the standard offer. So I did the, my first of many Steve conversations which was the classic, let's walk around the building. He hated to talk to people in his office in a standard

  

setting, his thing was, let's get out and, so we made, you know, five or six laps around Bandley One and we chatted about a bunch of stuff and I guess he thought I was an okay guy. Steve and I stayed in touch and he, and when I went to work for him he was, he came to appreciate my kind of organization stuff or let's solve problems. I remember I used VisiCalc from the moment it was out, and I created a table in VisiCalc of text, a status report. So at that point most of the status reports he got were, "I worked real hard last week and I took 40 phone calls" and stuff, and so what I created was a report on the Apple II that, in one column had all of the big issues, and then it was like who was working on it and what was the resolution or something like that, and so I basically used the, this VisiCalc as a word processor, but you couldn't do columns and certainly couldn't do tables in the Apple II word processor at the time. So when I printed this out for him he said, "That's it, that's what I want," because he was also a what's the most important thing, let's get it done kind of guy right?

So we got along fine until we conflicted and what we conflicted over was the fact that I'm writing millions of dollars of checks every month to the Lisa and Mac groups, and I'm getting no resources to, I mean, we were running the Apple II, Apple III division, R&D on four percent of sales. I was in rivalry with the software group within the Apple II division for who was going to be most profitable. The Apple II went out the door typically between 60 and 70 gross points of margin. Unheard of for hardware right? And that money was going straight to these development groups and I said you can't keep this up, when I tired to do the IIe I almost got, damn near got fired in that, and finally Woz and a couple of other people intervened and said no we really probably need to do that improvement, 'cause Steve wanted to cancel the IIe. And actually probably the only thing that saved it was the, Broedner's ASIC chip right? They did this incredible chip where they put most of the Apple II logic on a single chip and that was sufficiently interesting enough technically, you know, and fit the mantra of fewer chips, so that Steve finally I think went along with it.

But he really, I once said Steve was like the, his vision of a half a billion dollar company was a company that was built on the ruins of the previous half billion dollar company. It's like the old, well think about it, you know medieval cathedrals, they're almost always built using the stones from the previous church right, because it's a lot easier to do that than it is drag stones from another place. And so, my notion was well, you know, we could probably have a billion dollar company if we took the old half a billion dollar company and then built a new one next door, right? <laughs> And, so you didn't really see eye to eye, for him the old was a rival of the new and threatened the survival of the new, right? I remember one of my tasks as Apple II product marketing guy and then the Apple III group was making the rounds of engineering to cheer up people that Steve had been visiting. So when he was recruiting people for the Mac group his notion was of course they're the best and the brightest and everybody else is sturm und drang. And he would wander into somebody's cube over in Bandley 3 and say well, you know at this point he's the Chairman right? And not to mention the icon that Scully was trying to make him into later on. And so he would come in and say "Well what are you doing?" And the engineer would go, "Oh, well, I'm working on this bug, you know we have this really tricky problem that, you know, if this happens and stuff like that, and I think I've just about solved it." And Steve would go, "What? You're working on a bug, you're fixing this old crap when you could be, you know, doing great things blah, blah and stuff like that, you're just worthless aren't you?" And of course, then he would leave. And of course the guy's in his

  

cube vibrating and like, you know, so much Jello, and I'd have to come by and explain to him that, yes, fixing bugs and helping customers was really an important thing that the company needed to do.

But on the other hand, that ability to not care about the impression left on other people was incredibly powerful, I mean it was like a drug. But on the other hand it got him into incredible problems. I'll never forget, I got a call from Jean Richardson who was Apple's Director of Marketing at the time, and she said you got to get over here right away, we've got these visitors in and you have to talk to them. So I came over and I said what's going on? Well, I turned out there was a delegation from Japan and, I don't know if this story has ever been told that many times but, they were writing a series of books for Japanese schoolchildren, kind of a Horatio Alger series.

**Brock:** Okay.

**Pohlman:** And they'd decided that the one American they were going to feature in this series was Steve Jobs. Mr. Innovation right? So they had done all the research for the book and had it all set and, but the only thing left remaining-- Now some of this may be a bit apocryphal so, I'm doing this from memory but, this'll be like '80, '82, so they sent a delegation over, put them in a conference room and Steve is supposed to come over and talk to them and they've got their tape recorders out and running and stuff like that so all they need to do is get this final interview and they'll put the wrapper on the book. Well Steve waltzed in, according to Jean, about ten or 15 minutes late, and proceeded to chew them out. "What are you thinking? Japanese innovation, another fish flopping up on our shore, you can't possibly know anything about innovation", yada, yada, for about five or ten minutes straight out, turns around and walks out. Tape recorder's still going. You know and they're all looking around I'm sure for swords right, to do the-- And, so Jean basically said that's what happened, you need to go and like tell them the history of Apple or do something, get them out of here, right, exactly. So I went in and did, you know, ten minutes or 15 minutes, I don't even, I have no idea what I said, but it was just designed to make them stop quivering. And then they left and the book never was published.

But that was powerful, it was like, how could you do that? And the answer is, I don't know but there's a certain fascination about it. I was quoted by the [San Jose] Mercury News when Steve was exited in '85 as saying that working for Steve was like riding backwards on a roller-coaster, you weren't sure where you were going, but you were definitely getting there in a hurry. And that was the feeling. If he believed in you, you could crash though walls, right? If he didn't believe in you, you know you didn't have a life left. There's a lot of cult thing about that. I mean you go back to the Joneses of the world and so forth and it's that singular belief in a powerful leader that has no fear, and no doubt. If he did have doubts about anything I never saw him express them.

**Brock:** Interesting.

**Pohlman:** And again, that's powerful, because most people are plagued by doubt, "Oh am I doing the right thing? Oh is this the right thing to do? Oh my God what's gonna happen? No, that's wrong, that's right." Now the thing is that that was part of his power too. I said, he was not a technologist, but you put him in a room with 20 products, and he'll spot the two hot products every time. He had an uncanny ability

  

to say "That's trash, that's not." But, if you put him in a room with a white board and give him, you know, a marker, and say come up with something hot, not a clue.

**Brock:** But exquisite taste.

**Pohlman:** Exquisite taste and a great editor. Right?

**Brock:** And was Steve Wozniak working, was he that three percent that you were able to spend on?

**Pohlman:** You put Woz in a room with a bunch of parts and he would invent something.

**Brock:** Right.

**Pohlman:** Right? So in that sense, yeah, so. So Woz was, he was interested in stuff, he, you know when he left Apple the second or third time or maybe after the whole "Rocky" episode at Berkeley, you know he started Cloud 9 and one of its first products was a universal controller for, you know, TVs and electrical appliances and stuff like that, and he cooked up this idea that a microprocessor could be general purpose programmed to control all the, where the heck did that idea come from? You know, it was just one of a gazillion kind of things.

And he was completely, just completely genuine. He was the only person I knew at Apple who never changed from the first day I met him to practically now. I mean, it was that same open personality and, you know, I remember when I was first at Apple I was on a bowling team with him, Apple had a couple of teams in a league and stuff like that, and he was just a very genuine person. And he was also kind and sensitive. I remember Woz set up a, just before the company went public, one of the things that they noticed was they hadn't exactly been, because of this whole HR mangle at the beginning, they hadn't exactly been equitable about the passing out of options. And in fact they had what a lot of us called the "guilt round" just before the company went public where they gave a bunch of people stock that they had just kind of, you know, accidentally screwed over, because each manager sort of decided on their own, so some people were very generous. People who worked for Couch did extremely well, people who worked for others did very poorly and, so they had the guilt round but Woz set up a table outside Bandley Three, and sold people founder's stock at seven bucks a share I think it was. And this was literally months before the company went public, the company went public at $22. So people lined up and just bought shares, it was founder stock, it had to be cleaned so it wasn't exactly liquid, you know, from the get-go. But he just felt that, he realized people had gotten screwed and he just decided to do something about it.

**Brock:** He did his own employee stock plan.

**Pohlman:** He did his own employee stock plan. And the smart employees lined up at that table. But yeah, that was Woz.

**Brock:** Well, maybe we could move then to discussing how the ideas for leaving and for Forethought come together. Discussions with Rob Campbell.

**Pohlman:** Yeah.

  

**Brock:** What was the germ of it? How did it get--

**Pohlman:** Well, there're actually four of us that started the summer of '82. All four of us within the Apple, in those days called PCS, the personal computer division, as opposed to the Lisa or Mac divisions.

**Brock:** Okay.

**Pohlman:** It was Rob, myself, Mike Kane, and-- who was Apple's Special Delivery software first. He was an Intel guy, but a software guy. He's still in the Valley. And a guy named Mike Connor, who was at that point the Apple II product manager. And we were trying to figure out how to put a company together, and what did we want to do? And we looked at every business model you could imagine. One of my favorite was we talked about forming a guild. A software guild. And actually hardware. So the idea was we'd build a big-- we'd take a building, and we'd do some labs and we'd have a gazillion electronic parts and all this kind of stuff and a bunch of software development systems, and we'd basically just attract people. Like Hertzfeld and people like that. Like Bill Budge. And you could be a member of the guild, but the way you had to contribute was you had to do something, you had to develop something. And the guild got the rights to that, or part of the rights to it, and that's how it funded itself. So it was just going to be a big sandbox. And I still kind of wish <laughs> we'd done that. But we couldn't-- we found-- and remember, in those days, Apple was pretty loosey-goosey about people doing stuff on the side. So I remember… <laughs> In fact, I remember this funny story about Hertzfeld, so… And this relates to the Apple II. Remember, the Apple II didn't originally come with 80 columns text, right? So there were a number of Apple II, 80-column boards. And finally Apple decided we wanted to do our own. So I happened to know that Andy Hertzfeld had written the code for the 80-column board that-- I forget who put it out now, but it was the original one, very complicated piece of hardware, but decent software. I went to Andy, and course, he had been an Apple employee through that whole process, right? And I said, "Andy, we really need you to write the code for this 80-column board we're doing." And Andy said, "Well, you know, I don't really think that'd be ethical, because I did the code for what's his name's board."

<laughter>

**Pohlman:** I said, "Andy, you did that code when you were an employee and you didn't get a release, and you just did it. And now I'm asking you to do it for the company that you work for, that we're paying you, and you're saying you can't do that because it's unethical. Okay. Fine." But that was Andy. But, I mean, he had a moral sense. It just was a little bit strange. But there were a lot of other people that did side projects. I remember a guy that, the guy who did Apple Plot deeply resisted the 16-sector DOS that we were going to put on the Apple II, because he felt that would cut into his sales of Apple Plot. Now, he had been an employee through this whole process too. In fact, he ran Apple Japan, and… But he was basically saying, "No. I don't really want you to do a product because it's going to cut into my sales." Okay. And of course, Don Williams is famous, rest his soul, for doing software at the same time he's employed by the company, and then going out and promoting it to dealers. But… <laughs> So the company was a little bit loose in those days, but anyway, that's the story.

**Brock:** So there was the guild idea.

  

**Pohlman:** There was the guild idea. Then we started toying-- Kane and Connor kind of dropped out and decided they wanted to stick around. So Rob and I started thinking about stuff. I went ahead and left. Rob was still there. He was collecting a last option, I think. And at the time I had been working on the, like I said, I'd been working on the Apple IV, which didn't ever see the light of day. And then I started looking at this card for the Apple III, the 68000 card that Wendell had worked on. Did some simulation of that. And was just casting around. And then what we found, <laughs> what we ran across, was Dick Kramlich [New Enterprise Associates] and Phil Lamoreaux, of Lamoreaux and Glenn at that point, who were basically looking to do a software startup. And they had-- or remember, Kramlich had done, oh, PFS [Software Publishing Corporation], Personal Software, he'd done. And he'd just made a ton of money with Metcalfe on 3Com, going public. And they saw an opportunity. And we had some ideas about software. And as I said, I'd given up on the idea of starting a hardware company. I would've loved to have done the Apple IV, but I just, I knew what kind of investment it took and it just didn't make any sense.

So we started playing around with these notions of the software. And I started a business plan, putting a business plan together about object-oriented software and what kind of leap we could make relative to the software interface. Because we were looking at even what was going to be on Lisa and Mac. And as I said, I really felt like the compound document approach was going to be important. I really felt like that we could build a platform for that software, because the real notion and what started Forethought was this idea of an environment, we called it Foundation, that was going to be a place where people could put object-oriented code. It was going to be an extensible, extensible platform.

So if you think about compound document approach, everything, in a sense, can be subsumed in the notion of a document. So its output of some kind. So a set of table, a database, is simply a set of tables. Visually, it's a table. Or a series of tables. But at the end of the day you want to output that to something as well, and so you need to put it into a document kind of environment. And the idea was, "Well, what if a database was actually in effect subsumed in a document." So we could, for example-- actually, I'll skip to the Oracle part later, but… Because it's a fascinating sorry there with Ellison. But we start put this plan together, and what we could do as a company. And the investors were interested, I think, because they hadn't seen anything like it. They were very aware of Lisa. They were somewhat aware of Mac. At that point, PFS [Software Publishing Corporation] was in on the Mac idea. This would've been early '83, right? So they'd been disclosed. Microsoft had already announced the first Windows. VisiCorp was out with their thing. So it looked like this was trending, but nobody had done anything yet.

So they-- and the other thing too was that Rob and I had, at the time, a reputation. So Rob had done software for a long time. There weren't very many people who had five years of software experience in 1983, right? Or in Pat's and Rob's case, probably six. And I had plenty of history, plus I'd done the turnaround. I'd done the [Apple] II. I'd done the turnaround of [Apple] III. At that point, things were still perking up. And so we were probably bankable, at least to start with. And we had a few bucks of our own. So we threw some money together, Rob and I threw some money together, started the company, and then we put almost immediately a first round in. And that was all. I wouldn't call it a clean sheet of paper round. It was $600,000 from, split between Lamoreaux and Glenn, and Kramlich's New Enterprise

  

Associates. But it was basically money to write a plan and hire some people. And so that's what we started to do.

**Brock:** Was there an example of this type of environment and these kind of live objects?

**Pohlman:** No.

**Brock:** Is Smalltalk at all like that or any of the stuff at PARC?

**Pohlman:** As a programming environment, yes. But not as a business environment. Remember, our notion was that you could build applications with this. I mean, if you had databases and you could connect them with logic and stuff like that, then you could create workflows. I mean, there was a lot to the notion of live object interaction that we thought was exploitable. So we saw it as a, we called it a business-- actually, it was interesting. We had talked to Budge. You know, Bill had, at that point, had done something called Pinball Construction kit, one of the most brilliant pieces of software ever written. Pinball Construction Kit was an Apple II application. And what it had was a toolset. It was kind of object-oriented. It had a toolset of levers and flips and pin and ball and dials and accumulators and stuff like that, all on a little tray. And you grabbed a flipper and you stuck it on the screen and you grabbed another flipper and stuck it on the screen. Then you had a thing and it would shoot balls. And you could build. Because Bill got tired of building games, he said, "Well, let me just build a construction kit for games." And it had gravity. I mean, he was enough of a, he understood physics well enough, so that the ball was realistic. It would go up and it would fall as you would expect it to fall on a gravitational field. So the pinball construction kit we thought was brilliant. And we'd first talked to Bill about, he was living up in Piedmont at the time. We first talked to Bill about the Guild, we saw that there was a business equivalent of the pinball construction kit. What if we could build business applications with different objects <laughs> in the tray and we literally, part of that early business plan, was what we called the business, the application construction kit.

And so the notion was if we could hook and pipe things together, we could build applications. And in fact, we tried to hire Bill to help us build that. Because we thought he-- but he was not-- at the time. Ah. At the time, he had a couple of deals with Nintendo and he was working on, he was trying to finish up an arcade game, and he just didn't have the time. But he was kind of interested in the idea, as a different place to go in his career. So that was our notion. And we felt like if we built that kind of environment we could actually have some developers in the community, not just our own. Not just building an Apple writer, or a MacWrite or something, but actually build something that people could use as a construction kit.

**Brock:** After you had these notions, this kind of conceptualization of what you would go for, did you then go and hire these, the two people that you mentioned from PARC. Did then you go looking for people who had--

**Pohlman:** Mm-hm. Well, we looked for--

**Brock:** --the skills?

  

**Pohlman:** We found them, and then they found other people. So it started with-- I want to say it started with Bishop, who knew Rowell. So Bishop had done the file system for the [Xerox] Star. Brilliant guy. And he had then gone to National to do the object-oriented programming environment on the National 16000 chip. Remember, that was-- oh, shoot. Pierre Lammond's baby, Mr. CMOS, at National, had. And they envisioned themselves competing with Motorola. So the 16000 was a 68000-class processor. And it needed a development environment. So they hired all these guys, including Bishop, who knew Unix really well. They built a Unix environment. They built a, they started building out a development environment for it and so forth. And then when the plug got pulled on that they were casting around and we found Bishop.

Bishop knew Rowell because Rowell had worked at PARC with BravoX and done the editor for that, or at least worked on that editor. Brilliant guy. As I said. Does most of his work now in search, if he's still active. It's a shame. I remember… <laughs> Remember meeting with him in the late, I want to say, early '90s, and he was trying to explain this company Google to me. And he said, "You know, it's a great search engine. It's the best search out there." And I kind of went, "Oh, yeah, yeah, sure, Peter." But any case, so Rowell came in and Rowell was working on a-- he was working on this two-byte editor. So he was trying to build a kanji editor. And he was doing it as an object-oriented editor, because he wanted to do two-byte codes and he thought that was the most elegant way to do it.

So we never brought him in as a full-time employee, but what we did was we had, he was under contract, to build the object-oriented editing platform. Because we could take his two-byte character set object editing and put more in different objects within it. So the notion was it was going to be a container for editing all these different objects or at least dispatching editing routines to them. And then Bishop was working on the structure of Foundation, but also on the database notion. So our idea was we had to put a database platform under this. And at the time when we looked, and again, this is '83, it looked to us like SQL was going to be the way to go. And the only really viable product we could get our hands on at that point was Oracle.

Now, Larry [Ellison] had maybe 60 people in the company at that point. Had gotten-- I'm not even sure he got, had, any investment at that point. I think it was pretty much self-funding, and… But we looked at all the relational databases at the time. And settled on Oracle. And struck a deal with Larry and what was his name? Umang Gupta, who later went on to do Gupta Technologies. So Umang was his PC guy. He was trying to do a PC version of it. And we cut a contract to make Oracle the underlying database technology in Foundation. We struck a very interesting deal with them.

They had no ambitions at all about a graphics-oriented interface on a database at the time. And so they gave us the rights to that for the PC and we had three different royalty schemes. One was a basic, for basic foundation, where the tables had to be predefined and the user had no ability to create a database with it. So it was just going to be a container for our stuff. And that was at, I don't know, think it's 25 bucks a copy or something like that. And then up from there were all the way to full relational access, building multiple tables, doing joins, the whole shot. So that was going to be the underlying database technology. We had Rowell's editor for that. And then we hired a guy out of Atari who'd done-- he'd worked on some of the early animation in the Superman series. The first Superman had some kind of animation. So he'd been used to big color graphics animation systems. He was doing some of the

  

graphic stuff. We had a guy, had a couple of other guys we hired, who were basically building. We hired a guy also out of the National group that did the internal linker and loader. I mean, we had to build a whole environment in this thing. Because remember, in those days, compilers for the Intel chips, that's before the 286 showed up, we had to build middle model compilers to even get past 65K limits on code and stuff like that. And then, of course, there was the whole C/C++ problem. So we were basically-- in fact, one of the things.

Rowell wrote a debugger for C at the time, and we started to productize that. One of the notions was because we were going to have a long time building this environment that we could maybe sell our development tools as we went along. So we had our middle model compiler written and we were going to productize that. We had CDB, it was called. It was the most popular C debugger. And that was going to be a product and so forth. So, I mean, we were launching along and then we raised-- we had that plan, we had those people on board. Was a pretty impressive group of people. The Mac was about to be launched, and we put a couple million in the bank in December of '84, and that was not only the same group of investors but also they brought in Abingworth Limited and a couple of other of the follow-on to New Enterprise partners. And then that got us into '84, when it hit the fan.

**Brock:** <laughs> Well, before we get to the fan-hitting, it was interesting to me that right from the very beginning there was the idea that you could develop and sell software along the way while you were creating this kind of larger project. But I've wanted to ask a little bit about the sort of computer systems and software that those people were using to develop Foundation. And also the type of hardware that you envisioned Foundation eventually going onto.

**Pohlman:** Right. So we called the company Forethought, which investors loved the name, because, and I used to say, in the software business you have to shoot ahead of the duck. The nice thing about hardware was it's predictable. So we could tell where it was going. You could see the chip being developed. It was going to go into a system. It was going to have some memory. And so the Moore's Law trajectory was pretty obvious. And not only that, the timing was even fairly predictable. Software, completely unpredictable. So the only thing you know about software is you have no idea when it's going to ship, right? But the hardware was pretty regular and we could see what was happening there.

So our notion was if we start developing software today on today's hardware platform, we'll always be behind. So why don't we develop software today for the hardware that's going to appear in two years? The good news is that that hardware kind of existed, it was just insanely expensive and it wasn't being sold to the right market. So for example, our early develop environments looked at Sun. You know, Sun box was shipping in '83 and it had a Unix on it and so forth. And it had a graphical, I mean, it's sort of, I mean, you know, it had a bitmap display but it wasn't, the software wasn't there. You know, the applications weren't there. But we thought, "Well, maybe we can do that." We looked at the Lisa as a place to maybe prototype, because it wasn't selling in volume but at least it had the right kind of hardware.

In reality, what we wound up with, particularly with Bishop and Rowell, is we had a VAX. We had a 750. And the compilers. And we were basically doing part of the environment that we invented was this cross debugger. So we could basically, our target machine, let's say, at that point, was an IBM XT. We could

  

actually debug the software via the VAX, because it, I mean… And literally getting Foundation up. And we build Foundation. Getting it up was literally, you know, a line of code <laughs> at a time almost from a debugging standpoint. So you couldn't debug on the same machine you were doing the development on. So we needed that kind of remote debugging capability. But… And so actually it was Rob that financed that. I think we leased it through him. But it was a powerful enough machine to serve multiple developers at the same time doing code, and we had Intel cross compilers, so we could compile stuff on that.

Frankly, it was a bad decision in the following sense. Had we looked at both Apollo and Sun as development platforms, had we done that, we would've probably wound up with code that would've run on those platforms, and therefore might have had at least a small initial market. But the thing we guessed wrong about was the amount of time it was going to take to get that hardware out. So when the Lisa failed, we had no sales platform there. The Mac was too small. It wasn't going to, it was never going to work. Apple didn't do what we thought they were going to do, which was a next generation Mac soon enough that had the right capabilities.

And IBM took forever to get the 286 machine out. The XT ultimately was too slow for what we wanted to do. The 286 machine took a long time to get it out, and basically, Windows was a failure. I mean, it didn't work. But they managed to kill VisiCorp in the process. So we had no-- and we really didn't have the low-level graphics routines that we were expecting to have. So we wound up with a product concept, some initial code that ran slowly on the XT, but no real volume platform that was fast enough and cool enough to do the job for us. And that, that was sort of a-- oh. And then, of course, Oracle told us in,, January, February of '84 that instead of shipping the code in April or May of '84 it was probably going to be April or May of '85. And so we had no database platform. And we wound up in a lawsuit with Oracle. I remember, you know, I got deposed, Ellison got deposed, Umang got deposed. It was a real bloody mess. Which didn't win us any favors with the investors either. I mean, all they wanted to do was invest in Oracle.

<laughter>

**Pohlman:** And here their investment was basically suing Oracle, right? We settled, eventually, and they never admitted wrongdoing and so forth and so on. But basically it left us with no way to build the ultimate platform and frankly, no real volume hardware target to put it on.

**Brock:** And was that, that was all coming into, that problem with there would be no place to put it, <laughs> that was emerging in '84?

**Pohlman:** Yeah.

**Brock:** Okay.

**Pohlman:** Yeah.

**Brock:** You could see that?

  

**Pohlman:** Remember-- yeah. Because it was-- by ’84, you know, course, Mac was launched in January of that year, it was, from what was on the drawing board, there was nothing behind it, really. Lisa was failing. It was too expensive and too slow. And there wasn’t, the Lisa II that they were talking about doing, wasn’t happening.

And again, we expected the 286 machine to ship in ’84 and it didn’t. I don’t think it shipped until, let’s see, ’85 maybe or something like that. I can’t remember. But any case, things weren’t happening. And what we had was the XT and that was not going to do the job. So… And with the Oracle problem and with the fact that Foundation was-- we had a version of Foundation up and running that could do multiple fonts, could do tables inside and could do bitmap editing. So you could draw in the middle of a Word document. We had that working slowly, but at least we proved it. But at that point the investors didn’t really see a market for it. And that’s when we changed strategies.

**Brock:** What was, in ’83, well, what was your specific role? What did you find yourself doing most of all?

**Pohlman:** Well, I was president.

**Brock:** Yeah.

**Pohlman:** But mostly it was engineering. And so I was, of the two of us, I had the most engineering experience and the, probably the stronger vision of the object-oriented architecture. So I spent my time with Bishop and with the other guys. I hired [Brian] Sterns out of Apple and he wound up going back, and hired Rick Aricchio, who you may remember. Rick worked for us for a while. I hired Carol out of documents to put-- our idea of manuals in those days was 50 pages max, and this was a day when applications came with things like that [indicates huge pile of pages] We said, “If it isn’t obvious to the user…” There’s a famous quote, actually. Here’s a guy. If you haven’t interviewed him, you should. Stuart Greene. Interview him? He was a disciple of, well, that whole group up in Santa Cruz that did mouse type stuff and so forth [Doug Englebart]. But Stuart Greene had a phrase he called, for software design, he called the “principle of least astonishment”.

**Brock:** <laughs>

**Pohlman:** So the notion is how you think it should work is the way it should work.

**Brock:** <laughs>

**Pohlman:** So… And it’s interesting. If you drive today’s cars, right, you confront that principle all the time. So I poke at the radio. <laughs> And what I’m expecting is a tuning knob or something that’s similar to it, or a slider or something. And what I find in some cars, because I rent a lot of different cars, is that there’s some cars I can’t even figure out how to work the radio. And it’s like, “Who designed this interface?” Because the principle of least astonishment says is the first thing I should try in order to do something, is the thing that should work. And if it takes me four or five different tries to get something to work, then it’s a bad design. So the principle of least astonishment was what we were looking for, is-- and the notion in Foundation was if you poke something, it will tell you what it can do. Right? You don’t have to sort of

  

look up a manual and say, “What can I do to a paragraph?” You just poke it and it’ll tell you. And so that’s-- yeah. That’s the essence of what we were trying to do.

**Brock:** And did you have a small board at the time? Was that a group that you were interacting with a lot? Maybe you could talk about your board and…

**Pohlman:** Well, our whole thing, we hired this brilliant PR person, and her message was, “Everybody’s talking about what they’re going to do and they never deliver on time. So your secret is going to be to not talk about what you’re doing.” Everybody wants to know what you’re up to. And particularly when Metcalfe joined the board. So I recruited Bob, and it was the first outside board he’d sat on. And one of the reasons for doing that is in addition to him being a prince of a guy and a life-long friend is that he had made much money for Dick Kramlich. And his credibility with Dick was incredible. And he’d been at PARC, so he knew what the heck this was all about. He’s one of the few board members you can imagine having that actually understood what we were trying to do. And had an appreciation for it. And was enough of an engineer certainly to know where the heck we were in the process and to respect the other folks we had.

So there was a lot of excitement about what the hell were these guys up to? And we didn’t tell anybody. In fact, we had a launch party, when Bob joined the board, and a big celebration. Invited all the press and so forth and so on. And I remember standing up and thanking everybody for being there and they were all waiting for the big pitch. And I didn’t make it. I just said, “Thanks for being here.” And we were attacked by the press. “What are you guys doing? What are you up to? What’s your product?” so forth, so on. “Nah, not going to tell you.”

**Brock:** <laughs>

**Pohlman:** Drove them nuts. And so when we actually finally did launch… So we never launched Foundation. Which is probably good, because we never were going to ship it, or never did ship it. When we finally did have a launch and we attracted a lot of attention, a lot of press, because nobody knew what we were up to. And so that part was fun. The board liked that. They liked the notion of what we were doing. They liked the fact that they had a relatively small investment in a category that they thought was going. And remember, at this time, a lot of VCs weren’t touching software. I mean, we talked to Valentine and his folks. And in fact, I think at the time I want to say Pierre [Lammond] was working for Don’s firm. And Pierre wanted to do an investment in this and he couldn’t talk Don [into it]. And Don was more like-- he liked fabs, right?

<laughter>

**Pohlman:** From his perspective it’s like, “Well, let me get this straight. You’re going to sell these black squares and they cost a buck and you’re going to charge 100 bucks, but what’s their scrap value again?” And he just, <laughs> he just didn’t get software. And that’s okay, but… And there weren’t that many firms that really did, particularly at the kind of software we’re trying to do. And so there was this quiet excitement. We had a fairly small board. It was the investors, Rob and myself, and Bob as the outside board member at that point. And that was the right mix.

  

**Brock:** When did it become, or how did the change of plans come about, the realization that you were going to have to make an alternate plan when you realized Oracle gave you bad news, et cetera?

**Pohlman:** Well, yeah. What we did-- yeah. What we didn't quite understand, Rob and I, and certainly me, was how much the investors were counting on that Oracle relationship as an underpinning of the value of the company. So, because, again, I mean, they saw that as going to the moon. They couldn't, Larry [Ellison] wouldn't let them, invest. Because he was holding that really closely. So in that sense it was like Microsoft. I think Microsoft only had one outside investor before they went public. Or before they did Mezzanine.

And so when the Oracle deal hit the silk and they couldn't see a route to the Foundation we were imagining, and then we were all looking at the hardware platform issue, it was a crisis. And we were in danger. We were in danger of losing the company at that point, but we had, and so we had to replan really rapidly. And if you looked at the assets we had, we understood this market, we understood the PC market, we certainly understood graphics-oriented software. So that's when we created the MacWare concept. And looked at publishing instead of developing. So we said, "Hey, Foundation may happen. It's going to take a lot longer than we thought, and we don't know what we're going to do for a database in the meantime, so… But we know how to get stuff out the door," and there's no, there was no software for the Mac. And people didn't know how to write it. I mean, if you look at PFS file, for example, on the Mac, it was, I mean, I loved him like a brother, and John Page was an ex-HP guy that I'd worked with in GSD division, but they didn't know how to do that stuff. So they created kind of a piece of junk. And… But we did know how to do that.

And so we went to Apple and basically opened ourselves for business from a publishing standpoint. And those were, those were exciting times. I remember Aldus PageMaker. We saw a prototype version of that. Apple sent them over to us, because they were, he was-- we thought the notion, we called it in-house publishing. He had the notion of desktop publishing. And I remember looking at that product and going, "That's a primitive vision of Foundation." And in fact we thought it was too primitive. <laughs> We were pretty unimpressed, but… Shame on us. Course, we were unimpressed by other stuff that turned out to be pretty exciting too. But the point was we began to see a series of developers who had ideas. Again, PageMaker was, that was one of them. Then the Nashoba Systems, the guys were doing, had done Nutshell for the IBM PC. And they had this notion of a Mac database they brought to us. Rudy Dietzmann, who did a product called Factfinder, which was a quasi-object-oriented document system storing text for search and retrieval. So it was like a simple version of Macwrite, except you could have a bunch of Macwrites all organized with keywords and stuff. So we began to kind of connect with people. And since we had a decent reputation at Apple, all these people knew us from that. I remember talking to Tim Gill. We almost had, we had a number of products that we wanted to publish on the Mac, that Tim was thinking about writing. And he took Quark in a different direction, but… But yeah. So we talked to a lot of people. And we found a few things. And particularly we found FileMaker, which saved us. But the notion there was, "Let's apply everything we know about those interfaces to some code that, you know, where there's a existing development and a product that we can put to work and bridge that gap until we can do our own."

  

**Brock:** And that would be to help those developers refine their product before publishing it? Is that how you're bridging the gap?

**Pohlman:** Well, remember, yeah. Remember, the average-- yeah. The average developer… So you've got code. Now you have to put it in a sales channel, sell it, go through distribution all, and work with dealers. We knew how to do that too. We'd been doing it at Apple. We'd built the channel. I mean, when I say "we," I mean Apple Computer built that channel. And so for example, hmm-- I remember when we put FileMaker out, so we launched that in '85, shocking Microsoft, someone who had Microsoft File and they thought had exclusive, they had exclusive access to the market. We used the old Apple rep channel that was repping Macintosh or had been repping Macintosh and used them as our sales force. For a cut of the action. In fact, I remember that Mac had a [third party] hard drive almost from the beginning, and-- Hyperdrive, they called it. So it was a drive that you basically smushed into the back of a standard Mac case. We used FileMaker as the database so they could illustrate, so they could sell, more hard drives. Like, "Why would you want a hard drive?" "Well, you want to store a bunch of stuff." "Why do you want to store a bunch of stuff?" "You've got a database." So FileMaker didn't represent a huge opportunity in necessary itself but they could sell a lot of hard drives if they had something to show what a hard drive is useful for. So we knew how to get stuff to market.

The other thing too is that in the case of influencing development we also had the people who understood the interface and could coach them through a lot of decisions. I mean, I spent, I was sort of the internal product manager for FileMaker, and I spent a lot of time with those developers helping them refine that interface. And ultimately it was the prototype application for the Mac relative to how you would use an object-oriented interface to do forms and databases and the rest of the stuff. And a lot of ideas from Foundation that we had wound up essentially in, you know, in the interface to that product, although there was no code in common at all. That was all done by those guys in, you know, back in Mass.

**Brock:** So it was, you know, in the metaphor of publishing, you know, you were really being editors, developmental editing, of--

**Pohlman:** Sure.

**Brock:** --these programs before--

**Pohlman:** And naming it.

**Brock:** Yeah.

**Pohlman:** I mean, one of the most agonizing thing in software is finding a name.

**Brock:** <laughs>

**Pohlman:** Particularly, I remember, when we-- FileMaker is now, like, so well understood that you don't think about it. FileMaker. God, at the time it just seemed like the dumbest name on earth, but we tried every single thing we could with data and file <laughs> and everything, because trying to find something that already hadn't been taken and it was almost impossible. And in fact, even with FileMaker, we had a

  

conflict, it turned out there was some company that made metal files, but we managed to kind of get that cleared away. But, yeah so everything from naming it, to launching it.

Remember the other thing too is, in those days, the dealer channel and the customers were your market, but so was Apple. So there weren't very many Mac applications. And I remember Jim Hoyt who was with Apple from almost the beginning was running a support organization at the time, this would have been early '85, before the product launched. And I went to see Jim to demo FileMaker for the first time. And in five or 10 minutes, he just said "That's it," and basically the word went out, "There is a database for the Mac and it's faithful to the Mac -- and if you want to develop a Mac application, do it like this." And so, launching products, in fact Maxine Graham who worked for me at the time, and made a business out and launching products into Apple <laughs> for developers. Because that was an audience, if you could capture Apple's attention and you could get the attention of the field force, then you basically were 80 percent of the way there, and we used the same technique for eventually for PowerPoint. Because database, publishing and some of the early ads-- for example, in fact, I remember the ad agency that we used Filhart and Wright, right? Yeah, in San Jose. They had just lost the VisiCorp account, and they were really upset. And so, they signed us up as a way-- and they put all our best talent on our account even though we didn't deserve it, because they just wanted to stick it to VisiCorp. But Kurt Wright was our guy, and-- but the notion there, I mean it was really all about launch and we really understood that. That is now pretty much a routine kind of thing, but it was not well understood then.

**Brock:** Was it in this period, '85 after FileMaker that Apple invested in Forethought, was that later?

**Pohlman:** No, that happened later, and it was more about the-- if I remember it, that was really about getting PowerPoint out the door. What had happened is among other things when we hit the wall, it became pretty clear that we needed a real VP of engineering. And so, we looked around and talked to several headhunters and stuff, and through one of them found Bob Gaskins. Bob had been at Bell-Northern Research and had one of the early degrees from Berkeley in computer graphics. You know, the story's-- Berkeley in those days was a hotbed of-- you know, big hotbeds of software.

And so Bob came in, and his initial job was to assess for the investors, was to assess Foundation, how real was it? Was it the right track and so forth? And he agreed with that, and we retained the engineers that he had originally found at least for a while, but he also concurred that it was going to be a long time coming. And so Bob got that cleaned up, and put to bed and the-- started to be involved on the publishing side, but more importantly at that point, he was then already starting to talk about this notion of a product called "Presenter," which wound up being PowerPoint. And that was-- and we hired a couple of engineers, and worked on it and slowly crafted it. I still have some of the early design documents and stuff, as to what the look was supposed to be.

And so we had this parallel effort primary from a sales and revenue standpoint was publishing, and the sort of stealth thing was, it was PowerPoint. I remember in first looking at Presenter, or PowerPoint, and again, with this notion of the compound document, or object oriented architecture stuff, it was like "Bob that's nice, but it's pretty boring <laughs>." But he says "But I'm convinced. I'm convinced that a lot of people, if they just had something that would just do good presentations, that they would buy it." "Okay, okay <laughs>." And you know shame on me. I mean the thing that kills most companies is over ambition

  

and it almost killed us. In fact, yeah, we had several near-death experiences. The first one was actually before we shipped FileMaker. We had introduced Factfinder and a product called Typing Intrigue, that's best not mentioned any further. I mean it was a decent product, but we didn't know what the hell we were doing. But, we were basically-- we still had about a million and-a-half in the bank, the investors were getting pretty restless, they were very concerned about the Oracle thing, and so forth. And so, they were ready to pull the plug on the company, I want to say this is probably February of '85. And, we kind of figured "We've got to do something dramatic."

So we had Factfinder, it had been selling well through distribution, all by itself. We had done little bit of advertising, but we had not struck any distribution or resale deals on it. And so I went to, with our VP of marketing at the time, Darryl Boyle. We went to First Software and saw Mike, who was the CEO, and they had sold like four or five hundred copies in December timeframe of this product, and without any deal at all. So we said "We'll give you an exclusive for 90 days, but you've got to take a umptizillion copies, and you've got to give us a check when we ship FedEx, a check for the whole thing," a couple of thousand units. I remember he looked at me and said, "Have you got that many?" <laughs> I said "Yeah," but we did, we had a stack to the ceiling that we had built. And so, we went back, took the order, shipped it, notified him, the next day, the FedEx came, the check. Two days after that, the board had called a board meeting and Dick Kramlich, who's one of my favorite people in the whole world, who would never say anything bad about anybody, and would never try to break your heart, did not show up to that board meeting. He sent Woody Rae, who was a guy he used to tell you the bad news when he [Dick] didn't want to.

And so Woody was there basically to shut the company down, and take their money back and get sixty cents on the dollar, or whatever it was going to be. But we have the check from First Software for $140,000. And so I remember we have the meeting, "What's the status, how are sales looking?" "Well, sales are looking up Woody!" He said "Oh are they?" Yeah right. And I said "Yeah," and we slid the check across the table. And he said "Oh." I said "Yeah, we did an exclusive, they're going to be our distributor," you know, blah, blah, blah and so forth. He looked at the check, and there was some other conversation but I'll never forget, he got up, packed things up, instead of telling us we were out of business, and looking at the check he said, "You know, I wasn't really expecting anything so…tangible <laughs>." And we proceeded to cash the check, and the other thing that happened in that meeting, I believe it was that meeting was that Metcalfe had already seen FileMaker and he was convinced it was going to be a major breakthrough. And so he basically said publicly, and I think privately also, "Give these guys a couple of months, let's get FileMaker out the door, I think they'll-- you know we'll turn this thing around." And so between Woody's astonishment that we actually had a check and, Bob's support of FileMaker, we got through that window, and sure enough, FileMaker did launch, it did well and sales started to take off that fall.

**Brock:** I was interested in hearing more about when Bob Gaskins was suggesting that the company develop its own presentation software, and in addition to "Well, that's not I think what we were thinking about for our Foundations, that it's pretty straightforward." I mean eventually you said "Yes." So I was wondering kind of about the landscape of business-oriented applications, and how presentation fit into

  

like what was available at the time. How did it relate to, there were spreadsheets, there were graphing programs, and then was presentation an obvious next step, or was it a nonobvious next step?

**Pohlman:** I think it was-- well people were-- what is that saying "Any sufficiently large screwdriver can be used to drive nails," right? So people were brute-forcing presentations. We were starting to see it. Certainly, you could do something like that with PageMaker, and or even a word processor for that matter. You couldn't do-- I mean a lot of things like the slide sorter and the borders and things like that, they were presentation specific. It couldn't do-- but you could get some output. But what was really happening was-- and again, it goes to this Apple as an internal market. So Apple had done desktop publishing with PageMaker, and they had the rest of the-- they had the laser printer. Remember, ‘85 was the launch of the laser printer, so they needed applications that were going to show off that printer. That was one of the attractions actually they had for FileMaker, was that that could really make-- I mean it made sense, you wanted a laser printer to do the things that FileMaker could do for you, in terms of report generation, right? And graphics oriented report generation.

But Apple's marketing was getting kind of tired of flogging the same stuff. And so, when Forethought went to Apple with this notion of desktop presentations, it took fire. And remember, it took a couple of years to get Presenter out the door, as PowerPoint. But Apple saw it as the next desktop thing, right? And so you had marketing guys who had made their reputation with PageMaker who did desktop publishing, right? So they have made their reputations with this desktop thing, marketing campaign, from a thud-power standpoint, you had all this literature and all the rest of the stuff at the time, that you could do and make a name for yourself.

So there were a number of folks inside Apple that thought "Yes, this is the next desktop." And therefore-- and that was part of the-- whatever investment was happening was about giving Apple that window, and another opportunity to rationalize more laser printers, and rationalize its advantage because remember in that timeframe, you also had the Mac II coming out and so forth, and much more business focus. So this was going to be the next big thing, and that just fed right into where Apple wanted to go. And I think-- and as a new category, it was also fresh from their standpoint. We were desktop publishing and it in effect was a new category when it came out, because the alternative was, you know, Daisy wheel printers and stuff. And so, this notion that you could create published quality documents on your desk, was a competitive advantage that the PC could not match.

**Brock:** Was Microsoft looking at Presenter, before you left? Like you left Forethought in '86, is that correct?

**Pohlman:** Yeah, and the—right, and I went off to do another software company, and then eventually spun back into Apple in a completely different role. But yeah, what happened was that, it was kind of interesting. Ultimately my belief is that, and because I was still on the board for a lot of that time, but the-- and eventually left when I went back to Apple, but the key was that from a Microsoft standpoint, we had already kicked their butt once. I'll never forget that--

**Brock:** With FileMaker?

  

**Pohlman:** With FileMaker. I'll never forget that spring, we launched FileMaker at an Apple show in the Midwest and two booths over was Microsoft. There was another company, a forms company and they had this forms design. Well, it took me about 20 minutes to take their complex medical form and put it into FileMaker and have a real database behind it. So they were like-- you know, they were out of business. But two booths over was Microsoft with Microsoft File. And Gates had been talking for years about File and database technology that they wanted to do. And so Microsoft File was a fairly decent product, it certainly had a better graphics interface than PFS, they were convinced they were going to kill PFS in the marketplace, and basically they did. But they had no idea what we were up to, remember, we never announced it until we launched it. And so, here we are in booths and we've got a better product than they have. And everybody was amazed, because remember, FileMaker indexed every field, it kept track, it actually indexed every word in every text field, it was incredibly flexible. Microsoft File was fairly decent looking, but it was a classic database. You have a primary index field, and a couple of secondary's and blah, blah, blah. The report generation was not that great, certainly not as flexible as FileMaker. And, so they were pretty much astonished. I think their reps there were like, they came over "What the hell is that?" And we were surprised too, because we didn't know they were coming out with it, but we were pleasantly surprised since it wasn't as good.

So over time, over the course of a couple-- year and-a-half, two years there that FileMaker was on the market, we kept beating Microsoft File, until finally I think I want to say in '87 or '88, they gave up on it. But-- and they were also very tuned to Apple. And so here is Apple about to launch desktop presentations, now I actually have a copy of PowerPoint in the original Forethought packaging, even shrink wrapped. So, but I'm saving that for when I retire, and hopefully it will be worth a lot of money. But the point is that, we had the MacWare brand which we had launched FileMaker under, but we also had our own Forethought brand that we had saved for-- that we were going to do Foundation with.

So we launched, because PowerPoint obviously was internally developed, we launched it as a Forethought branded product. So we had it in the packaging, certainly Apple was all prepared for a giant launch activity, and so forth. And I truly believe at that point that Microsoft looked at it and said "Why would we let them do this?" They didn't have a competitive product at all, it was a new category Apple was going to put a ton of marketing behind. And they'd gone public and had a bunch of cash, and we hadn't and weren't going to, and didn't have a much of cash, and had some, what I would call some "tired investors," who had an opportunity to cash in and actually make some money. I mean, we had gone through at that point four rounds, five rounds, and understood the phrase "ratchet down clause" extremely well because our first round was at <laughs>-- our first round, our private round was two cents, then we did another one at 15 cents, then we did a round at 50 cents, and then next round we did at eight cents. That's when the ratchet downs kicked in. And so basically, we were getting diluted down, the investors were as well. And so, Microsoft made a cash offer at 35 cents a share, and it was good deal for the investors. Not a bad deal for us either, but that was-- and so that was the summer of '87, when they snagged it.

**Brock:** And you were on the board--

**Pohlman:** I was not on the board at that point, no. I was an interested shareholder. Rob was running the company at that point, and struck that deal. Rob and Bob went to Microsoft, kind of as-- Rob more in a

  

consulting role and Bob as an employee. And Microsoft actually struggled with what to do with it, I mean Rob did yeoman work to get them to continue to focus on it, and eventually made it successful. What really made it successful to me was being in a bundle, but-- and Microsoft would bring that together. It's interesting, I've often wondered if we had access to a word processor and to a spreadsheet that we were going to publish at Forethought, and didn't really have the money really, to launch the two of them, but it would always have been interesting to me if we had pulled off our own bundle at that point.

**Brock:** What had prompted you to leave earlier in '86, you left for a new opportunity to--

**Pohlman:** Yeah, I created a software company with a friend of mine. We had-- I still enjoyed software, the publishing was great, I loved FileMaker and as I said, I felt personally responsible for a lot of the interface, everything from some of the way it worked, to the notion of cloning a database, because I understood database and understood schemas most people didn't. And so I said "Hey, let's just copy the schema" and that really led to-- a lot of the software community gelled around FileMaker. Remember, that was a product that people launched careers with. I knew a guy who was in the-- believe it or not, the pool business. And so, he did spas and pools and had a pool service. What he found was that he had a knack for FileMaker applications. And so, he quit the pool business and launched himself as a FileMaker developer. There were hundreds of FileMaker developers, we got letters from them all the time. So, it built an ecology and I believe in that kind of stuff. And frankly, at that point, Rob was going fine and I had other stuff I wanted to do, and I also wanted to get back into development. And so, it was time, it was time for me to go. I mean they were-- I was able to keep the stock and so, I had an interest in the company and stayed on the board. But ultimately, it was-- I wanted to do Foundation, and when I really didn't have a chance to do that, I wanted to go off and do something else.

**Brock:** Right, and what was that?

**Pohlman:** Well, a custom software development company for a while. I got approached by a guy, believe it or not, a senior auditor at Bank of America. Fascinating guy, he had been taken out of Pakistan under armed guard, because he had discovered some government collusion with branches of Bank of America that was going to get him killed. So he was kind of famous within the bank. But, the bank had like 3,000 branches, and they only allowed auditors to be in their jobs for about-- internal auditors, to be in their jobs for about two years because they were concerned about collusion also. So they have this terrible problem in that, training time for auditors, by the time they really got them trained up, it was time for them to go.

And they had also, I think in '86 been hit by the feds on that $10,000 cash transaction thing, so their branches in Florida were just filling with cash and stuff. And so the feds asked them, "So how many $10,000 transactions that you have?" And they said "A lot." "And how many of them did you audit?" "Quite a few." But they had no worldwide system and they didn't have time to build one, and they didn't have time to train people on whatever they'd built. So this guy came to me through a friend of mine who actually used to work at Forethought, I had to lay her off, and she went to the bank in a support role. And so I got brought in, and I looked at the problem, and we built something called the Modular Audit System MAP, or Modular Audit Program.

  

What it was was, a Mac based application, the first and maybe the only one the Bank of America ever did. I mean, the number of antibodies around that was enormous. But basically what we did was, we created a program that securely would create an audit plan for a branch, or any other entity within the bank according to this set a set of standards, and allow the auditor to point and click, and build, and audit records of that branch. And then consolidate that up to a central thing so that they could report on every branch worldwide. And also interestingly enough, had a database that kept track of, if an auditor failed something and then changed his mind, you could replay the whole process. So, it even audited the auditors. But, so that became a contract, a friend of mine that was Apple's patent attorney, but also a closet coder came on board. We wrote the application, and we sold-- we struck a deal with the bank to split it 50/50, so we had 50 percent of the rights to the software in exchange for what we claimed was-- I think it was going to cost a couple of million bucks to do this, and we charged them a quarter of a million. And so we kept the rights and sold it to Bank Administration Institute and MAP was sold for years by BAI. So that got me back into custom development, then I got a call and we did that company for a while. Dan actually kept that company going for years, and then I went back to Apple.

**Brock:** Did they reach out to you to come back?

**Pohlman:** Yeah.

**Brock:** What was that like?

**Pohlman:** Well Apple was doing an internal skunkworks project, and they knew me as both a software guy and a marketing guy. It was called a-- not a lot of people know this story, it was called a "Fitness Center." So the notion of a Fitness Center is, that it's some place you go to work out and work on various things and stuff, but it's kind of a club. So Apple has the notion that they were going to do a club for Mac owners and it was going to have what turns out to be [the equivalent of ]a Genius Bar, and it was going to have software demonstrations, it was going to have support people, it was going to have everything but a cash register. Now remember, in '86, Mac was sold off, it was considered a “toy" right? So you know, real men bought IBM PCs, and DOS and whatever and the Mac was this little fringe thing that did graphics.

But Apple knew much better than that, but the problem is they had no way to prove it. So they were going to create these fitness centers, they were clubs, there was an annual membership, and your annual membership got you these services, and got you a place to go and play. And also got you a place-- and also created in the particular city, a place for the truth about Macintosh could be told. So "Is it a toy?" "No, go over there and see." And so, a way to kind of gather of the faithful, if you will. And, self-funding, and also a threat to the channel because it wouldn't take much to put a cash register in there, if the local dealers didn't shape up. Brilliant idea, created by a brilliant guy, Ron Rohner, who was Apple's channel guru forever. And, so I worked on that for a while, built their marketing program for small business and recruited a lot of software developers, and so forth. That got shot down and I wound up having to go-- I wound up spinning into Apple in a technical role, working for Michael Homer. Homer's-- he had a long history-- he recently died of terrible cancer, but he was in Google for a while, he did GO and all this other stuff. But Homer had built a group inside Apple marketing that did development. Dial-a-Node, which was the first remote connection to a network was done in that group. That group did the Lincoln Project, I don't know if you-- you probably heard rumors about the Apple-Apollo deal?

  

**Brock:** No.

**Pohlman:** No?

**Brock:** Well I'm sure Hansen [Hsu] has, but I haven't.

**Pohlman:** In '87, a deal with Apollo. Apple was going to port the Mac ROM to the Apollo system--

**Brock:** Forgive me, I have heard.

**Pohlman:** You have heard <laughs>? And they were going to put Apollo software on the Mac, Mac II at that point. There was one of the OEMs of Apollo that had signed up for hundred million dollars' worth of Mac IIs, if it could run Apollo software. A guy named Harry Chesley, who I think is in Microsoft now, was working for me on that project, and it was nearly done. Alan Oppenheimer who was at Apollo at the time wound up coming to Apple, worked on that. There was another woman who wound up at Apple who was on the Apollo side, and that was Apollo's low-end strategy against Sun. They did not have a low-end strategy against either Sun or HP. And so, the Mac II was it. I mean, after all they were all 68000 machines, the code-- I wouldn't call it a trivial problem, but it was not a big technical problem.

So, we built that inside this marketing group, right? Got rejected by engineering, there were a lot of rumors as to why, some people say John was trying to cut a deal with Sun. But for whatever happened, within a year of that of course HP bought Apollo, because Apollo was toast, without that deal they didn't have a survival path against Sun. So HP bought them which is why that stuff and that stuff and the 9000 Series and all that stuff was over at HP. So that kind of changed the whole landscape, there were a number of things like that that at that point, that kind of changed the world of computers kind of behind the scenes. My other favorite was the Mac board for the IBM PC, that was a Stewart Greene deal.

**Brock:** Now that's one that I have not heard of.

**Pohlman:** Yeah, well there were a number of Apple Fellows in those days and Greene was working on it and I worked on it a little bit, and the other Apple fellow was an ex-Apollo guy and he was the one that created it [Al Alcorn]. 68000 board plugged into an IBM XT. And remember, this was at the time when people were doing the transition between-- to the IBM PS Series, from the 286 machines, and so forth. So the 386 machines were out, IBM had announced that kind of crazy graphics thing and Windows were struggling and stuff like that. So our notion was, we were going to put this Mac board into a PC, it was going to be cheaper than upgrading the IBM PC itself. There was a huge base of XTs. The other thing it could do was, it could be a postscript driver, so you could connect a laser writer to the IBM PC, and have it work. There was actually a board you could buy for the PC that just did PostScript. And we were going to sell a zillion of them.

**Brock:** This was to turn an IBM machine into a Mac?

**Pohlman:** Into a Mac and run Mac software. So it was going to triple, quadruple maybe the base that Mac software could run on and it was going to blunt Microsoft and IBM strategies. Well, that didn't happen

  

either. Well, one is it was going to be ugly, right? I mean it wasn't a Mac, and there were some other reasons, but they're mostly religious.

**Brock:** Aesthetic?

**Pohlman:** Yeah, aesthetic, exactly. And, it was interesting because that was-- remember that contrast with the old Apple?

**Brock:** Yeah.

**Pohlman:** "Let's pay attention to what people want and kind of supply them with the tools to do it," and the religious thing that "The ROM is the crown jewels" I went-- I remember even approaching briefly Scully about it because what I saw was-- because one of the claims was, you know, "We've got a lot of different things to do in engineering, and it costs $21 million or some kind of thing, to ship-- you know, to create a product and ship it and we really can't allocate that at the time." And my message was, "Hey you give me the rights to this, and I'll get it funded tomorrow." Because I would have called Dick, and said "Hey Dick I've got the right to Mac ROM, do you think you've got a few bucks to support that?" And the answer was they would have written any kind of check that was necessary. And the engineering team was ready to go, I mean they basically prototyped the thing. So it's not that hard, remember that was in the days they were starting to build the chipsets and stuff, so doing one of the Mac-- smaller Macs-- was pretty straight forward. But yeah, so there was a lot of that religious war stuff. So the Apollo deal was all about religion, the Mac board was all about religion. And again, as I said, the sales was always the lagging indicator, so you look at those decisions in '87-'88, a look at where Apple was in the early '90s, there you go.

**Brock:** Was this also the period when Apple did allow like a third party manufacturer--

**Pohlman:** No, the cloning was after that.

**Brock:** Some years after this?

**Pohlman:** Yeah, some years after that, when it was too late. See, part of the problem was that remember, you had-- I remember being at a presentation to GE, to the CIO of GE. Now there's a guy with a big job right? And they were ready to buy a gazillions of Macs, I mean in '87-'88 there really wasn't a better choice for a machine that could do business, and where they didn't have to spend a fortune training people. Despite all of the fuss and everything else, that was going on, the business leaders knew that.

But they couldn't risk on a sole source, and so they were begging Apple, they'd say "Look, I don't care-- I mean, just appoint somebody, have him build 40 machines so I can-- you know, I'll buy all of your stuff, trust me I won't buy their stuff, just cut a deal with somebody so I'm not sole sourcing this technology." And Apple wouldn't do it. By the time they did, it was too late because Windows had caught up. Now remember, in '87 you had no-- I mean what choices did you have? I mean we looked-- Forethought looked at a port to Windows, it wasn't—Forethought tested every version of Windows that came out and

  

none of them were sufficient to support applications until, what? [Windows] 3. So and even that was pretty shaky. So, yeah, I mean--

**Brock:** I've never thought about that second sourcing aspect to the Mac.

**Pohlman:** Every corporation was basically saying, "Just prove you have a second source and the flood gates will open."

**Brock:** And I guess IBM, for their personal computer, they're IBM so maybe they don't need a second source?

**Pohlman:** Remember, it was Microsoft and pick your flavor hardware, right? From most people's standpoint, they didn't consider Windows-- they didn't consider Microsoft in the same sole-source class as because you could buy plenty of hardware that ran, and in fact the most important thing that happened to the market and since was Compaq, because that was a second source.

**Brock:** Yeah, right <laughs> inadvertently second source.

**Pohlman:** And in fact I remember, you remember that I told you when I left Apple the first time, one of the opportunities was to go help somebody build a better clone. So, there was a guy that approached me and he had some offshore money, and he was building a company in LA. And their business plan was going to be to build-- because they said "Compaq is 97 percent compatible, we want to build 100 percent compatible machine and beat Compaq." And said "You don't get it, software people are testing their software on the 97 percent compatible machine, so they can make sure they run on that. That 3 percent compatibility from their standpoint is risk, not opportunity." So he went away and I think he didn't start the company, hopefully he didn't. But, I mean think about it, I mean if you're going to look at a set of platforms, you're going to build to the lowest common denominator, because that's the widest market. Anyway, so yeah, I think when Apple hit that skid it was because of some of those decisions that did not take advantage of this-- what could have been an insurmountable lead.

**Brock:** And those, both the Lincoln Apollo effort and the Mac board for IBMs, those were both within this kind of technical development group within marketing?

**Pohlman:** The Lincoln thing was driven technically out of that, the Mac board was brought into that group, but it had been developed by-- gosh I can't remember the name. He was an Apple fellow, an ex-Atari guy, and I don't remember his name offhand, but brilliant hardware guy [Al Alcorn].

**Brock:** But that's where you worked?

**Pohlman:** Yeah. So I worked for Homer doing that. We did Dial-a-Node we did a bunch of other stuff. And then we ported a UNIX work alike to the Mac II, for a big government contract. We also supported, I remember was it Insignia? Or it was the outfit that did the first IBM emulator for the MAC, a UK-based outfit, and they came to us for porting support basically. And we helped them with that, and I helped them with some of their-- <laughs> they had a version, had it running on the Sun, and so they knew how to do the-- a 286 simulation on a 68000 machine and they wanted to price it for the Mac at like a thousand

  

bucks a copy. And I said, "No. I think you want to price it so that people just put it on the shelf in case they need it as opposed to, 'I desperately need it.'" And so they ended up coming out at a couple hundred bucks and did pretty well. And then, of course, ultimately there were a couple of other emulators that did better, but that was the first one. That would've been '87 probably.

**Brock:** And what was the Apple that you returned to? John Sculley was kind of in charge.

**Pohlman:** Right.

**Brock:** Full stop.

**Pohlman:** Full stop.

**Brock:** In this period of your return. Had the company changed much beyond the ouster of Steve Jobs?

**Pohlman:** No. Steve--

**Brock:** I mean, had there been other developments when you came back?

**Pohlman:** Yeah. It had. And not… Not in particularly good ways. John, John's a very sincere and a really nice guy. And I've known him for years. I still see him in Camden [Maine] from time to time. And I've always wished him the best. And he has one of the toughest lives of anybody I know. He still gets vitriolic emails, hate mail from people. Based on quote, unquote "what he did to Apple," or…

**Brock:** <laughs>

**Pohlman:** So I probably do fault John for two things. One is he threw gasoline on Jobs' ego and probably caused his exit from the company ultimately. I mean, remember, here's a guy that does have a very powerful ego. But John saw him as a marketing icon of Apple innovation. And so the bicycle campaign and so many of those other campaigns in the '80s that featured Steve. You know, when we, instead of a locomotive we have a thousand Volkswagens. So that whole series. What it did was, I mean, Markkula, Scottie, those guys, they knew how to keep Steve in line. I mean, it's like, "Look. You're just a kid. Shut up," you know. "And when you come up with a good idea, we'll know it and that's great." Right.

But he kind of went open-loop with John. Because John understood him as an icon. He was a smart marketing guy and he used him that way, but it wasn't good for Steve in the sense of his ego running open-loop. And John had some other issues with understanding, with evaluating people. He made a couple hires in marketing and sales that were not Apple people. Didn't understand the technology. I mean, he got people out of-- well, even the guy he hired to do Newton came out of consumer electronics and didn't understand the power that Larry [Tesler] had created in that box. Didn't know how to exploit it. That was my last job prospect before leaving Apple the second time, was I really wanted to be the product manager for Newton, because I had developed a handheld diagnostic tool in service. I was running, I was director out at the service at the time. I developed the first handheld diagnostic. It worked like, you know, the car plugin?

  

**Brock:** Yeah.

**Pohlman:** So you plug in a car and you get all these readouts. So there was a facility in the Mac ROM that could do readouts through the ports. Well, we built this handheld computer that could do that readout and print out a diagnostic on a machine. It's called TechTool. And so, I mean, that was the first handheld that Apple had ever built and I was very excited about Newton and I understood the technology. And all the data soup and all that other stuff was cool. And… But John talked Larry into hiring a consumer guy who didn't know how to get product out and stuff. And they wedged it into that inadequate platform like they did with the Mac. They repeated the same problem with the Mac that, with the Newton. And, well, the original Newton dictionary was, I don't know, 50,000 words, something like that. It was going to be a work slate. It was, Larry's design, was basically a tablet with a battery that had some built-in networking stuff. But it had enough memory and enough processor power. Remember, the standard in those days was MIPS per milliwatt. That was another Stuart Greene thing. So you measured performance in MIPS per milliwatt. Nobody knows what a MIP is anymore, because they've long since, you know, haven't seen a PDP, you know, PDP-11 in a long time. Or a VAX. But that notion of power and performance and memory, the whole handwriting recognition system of the Newton was designed with a certain sized dictionary in mind and a certain amount of processing power and memory. When they jammed it into the PDA format, they cut the size of the dictionary down and they cut the processor performance down. And the handwriting recognition didn't work. And remember when it first launched you would write on it and it would misspell things and you couldn't figure out what the heck you were talking about? The prototypes worked fantastic. But they were also <laughs> designed for a slate.

**Brock:** Interesting.

**Pohlman:** So anyway, I mean, I could go cranky, cranky, cranky.

**Brock:** <laughs>

**Pohlman:** In fact, when I left Apple the second time and the last time, I told somebody, I said, "The reason I'm going is I'm tired of standing around and being this naggy old woman who says, <impersonates> 'You kids are screwing up,'" you know. And I said, "It's time to just go do something completely different. Because it's not fulfilling." And that was essentially the first time too. It was just not fulfilling to do something that I knew was wrong, but couldn't do anything about. And if I hadn't switched to Apple Service and the support organization stuff, I would've probably left after the Lincoln and Mac board debacle for the same reason. But I switched careers.

**Brock:** Could you talk about that? Because, I mean, it's my impression that especially on the service side, and customer support side, well, that throughout I guess even the earlier period you were at Apple, but certainly in this period, had a high reputation relative to other makers of personal computers.

**Pohlman:** Well, first of all, it was about quality. We used to say, and that was essentially the genius of the Mac interface. You know, how do you do better support? You eliminate the need for it.

**Brock:** <laughs> Yeah. Best support call is one not made or something? Yeah.

  

**Pohlman:** I remember going-- well, here's the thing. If you look at that, there's a ratio that's critical to understand. Most people look at, "Well, how many calls did I take?" The real question is, "How many of those calls are problems versus how many are questions?" So if I have a question for you, that's of value. I want to know the answer, and if you give me a good answer, I'm happy as a clam. If I have a problem, then it's all about how fast do you solve it? And by the way, the echoing memory of the fact that your product screwed up and caused me problems. So let's look at that ratio, problems to questions. It was funny. <laughs> I remember when the first-- I was running Apple Service when the first PowerBooks came out. And they were built in such a way that, I mean, they were screwed together in order to save weight and everything like that. What we did in testing was, because we always did service readiness, we took 20 working units and took them apart and put them back together. Didn't touch them otherwise. A fair percentage of them failed.

<laughter>

**Pohlman:** And we went to engineering and said, "We're going to have to raise the warrantee charge on these things, because dealers can't repair these." That's why we actually wound up with a third-party depot thing, because the usual repair… I mean, Macs, the original Apple IIs, these, they were things that anybody could fix over the counter. Somebody would come in and it's broken, take it in the back. Little parts and stuff like that, and "bang," you're on your way. And Apple began to develop products that, where reliability was going to be the key. So we had a notion that most people say mean time between failure. We talked about annual failed rate. So set the target. So an annual fail rate says, "What percentage of a given product is going to fail per year?" So what you wanted was an annual failed rate of 10 percent or less. Because the average person held the computer for four to five years. So what you wanted is the average person to never have a failure.

**Brock:** Right.

**Pohlman:** And the notion there is if you have a failure, you think it's a fluke. And you just were unlucky. So it doesn't affect the perception of quality. If you have a higher failure rate than that, when you have a failure you presume it's a piece of junk. And if you don't have a failure, you think it's a fluke. Now, turning that perception around is enormous. Because there is a lag time in that too, right? The perception of quality, once it turns bad, it takes an enormous amount of good experiences for you to believe that you aren't just one of the lucky ones that, where it didn't fail, or conversely. So that focus was always there. The interface was built, so it didn't create problems. Remember the whole grayed-out menu thing was, "I can't make a mistake because it won't let me"? Right? And it doesn't fail because it's just built better. It was interesting. We had a challenge when Apple moved from 90-day warrant to 1-year warranty. For years the, you know, 90-day warranty was big, big industry, competitive problem. But the reality was that the failure rate really didn't-- there was no burn-in period or infant mortality rate in a Mac. They were burned in such that it didn't really cost us that much more to do that one-year warranty. We finally proved that to the company and got them to make the change. But that's, that part about quality, has been true for the company a long time. And the only, the only time the company kind of really stumbled was when they didn't live up to that reputation. And that was one of the reasons why we spent so much energy on the Apple III. Because the whole company reputation for quality was at stake. If we couldn't get that fixed, then the assumption was that they just ship junk. And that, that couldn't be allowed, I guess.

  

**Brock:** Was that part of that Hewlett Packard culture coming into Apple?

**Pohlman:** Maybe. But I think it was also basic empathy for users.

**Brock:** <laughs>

**Pohlman:** Remember, the interesting thing about Apple, and people forget this, is that the company, the reason the company was so messianic, if that's the right word, about the notion of personal computers, was it was the only company in the world where everybody had a personal computer. We knew what an enormous productivity gain personal computing was, because everybody had one. I mean, in those days, I hired, when I hired Maxine Graham into the Apple III group in '80-- what was it, '82, I guess, '81, '82, she came from Intel where they had one personal computer per department. If somebody wanted to do some work, they had to go borrow it or sign it out or something like that.

At Apple, not only did everybody have one at the desk, but remember, the employee purchase program, everybody got to buy one of their own at home. So it was very clear what the technology meant. And that, that created a culture that's, in effect, the employers are users. So it's personal. <laughs> You know. If it's a piece of junk, we know. You know, it's not like shipping bad vegetables to somebody, that you don't have to eat yourself. I mean, we were our own consumers. For example… And that was why, I mean, if I look at the things that Apple did that were real innovations in customer service. Take Service Source, for example. And so well before the internet, we shipped a CD-ROM with every service manual ever produced and exploded parts diagrams. And I could click-- this is in '91? I could click on a part on the exploded parts diagram and the Mac would issue an order to our parts factory to ship that replacement part to the dealership and track it all the way through. We used the GE communication system. Apple Order was the first. And they could use it to order finished goods also. That was way before the internet. Way before online ordering. But Apple, remember, essentially invented business email. The AppleLink system, which was available in '85, the way everybody communicated. So everybody had a computer and everybody could send email to everybody else.

**Brock:** In the firm.

**Pohlman:** In the firm. And to selected customers and dealers outside the firm. Because AppleLink was also available to Apple partners. So did we know how useful email was? Absolutely. Because we used it every single day, hundreds of times a day. And so when email became popular…

<laughter>

**Pohlman:** Later on, and businesses discovered email, it was like, "Duh," because the company ate its own dogfood.

**Brock:** That's, yeah, that's just the phrase--

**Pohlman:** Always.

  

**Brock:** --I was thinking of. I would be interested to hear-- could you tell, expand a little bit, on that AppleLink's-- I'm not familiar with it. And the Apple use of email.

**Pohlman:** Yeah. So AppleLink was a Mac application, and it basically talked to the GE communication, business communication system. And basically formatted in a graphics interface, you know, list of incoming messages and stuff like that. And, in fact, remember AOL, it had the first interactive interface. So they borrowed a lot of those technologies from the AppleLink product. But it was a software application. And because there was no, there were no browsers, there were no email packages and stuff. It was an application that ran on a Mac. Everybody had it. And used it for communication. People had AppleLink addresses similar to an email address, and kept lists and you could click and send things, copy a bunch of people and stuff like that. So it was used for communication worldwide. And again, built on this GE business communication system.

**Brock:** Was that a commercial product?

**Pohlman:** Yeah, was a commercial product.

**Brock:** Oh. Huh.

**Pohlman:** Yeah.

**Brock:** Fascinating.

**Pohlman:** It was used a lot. GE sold a lot of, oh, what do they call it? The electronic processing stuff where you could send orders and basically small communications about data transfer and so forth, but… So I forget exactly some of the underlying stuff, but yeah, it was all built on that. AOL picked that up and put the interface on it and then from there, once the web was there, it became obsolete. But what Apple did with building Apple Order was basically used that same communication system in a set of protocols to transfer order information, order status information and so forth, via the same network. And because we put the graphics interface on it with things like Support Navigator and Service Source. It basically changed the way that people looked at documentation.

We used to, I mean, the service documentation, it was a classic, right? You had binders full of stuff, and every month you'd update a binder and you'd send four pages of something that was supposed to replace page number 723, right? And you'd go into a dealership and you'd have the original binders here, and a stack of updates here. And they'd be calling on the phone going, "How do I do this?" Right. So Service Source was basically every month we ship the entire documentation set, all organized and in order and completely up-to-date, and just go look at that. I mean, this was at a point when CD-ROM was not, I mean, not even very decent interfaces, let along stuff like point-and-click for ordering and stuff, so yeah. In fact, the guy who built that in IT for me, that was my project on the service and marketing side. The guy who built that in IT was Pete Solvik who went on to fame at Cisco as their, as not only their IT guy, but the one who built the CCO, the whole online service system for Cisco. And then went on to glory as a venture capitalist.

**Brock:** Well, let's see. Do you want to just--

  

<off-topic conversation>

**Brock:** This has been absolutely fascinating for me. Thank you. Maybe before we go into the reasons that you, you know, your transition out of Apple for the second time--

<off-topic conversation>

**Hansen Hsu:** I was really interested in you had this idea for Foundation way back, very early. And I kept sort of thinking about similar things later on that Steve Jobs did at NeXT, which had an application kit, an object-oriented application kit.

**Pohlman:** Right.

**Hsu:** And then later on Apple was doing OpenDoc and then Microsoft's doing OLE and there was like Carbon, the workstation site. So was there any sort of cross-pollination going on? Like, was there--

**Pohlman:** Oh, they stole all our ideas.

<laughter>

**Pohlman:** No. I wouldn't say that the notions were obvious, but they became more obvious over time. So one of the things that we knew before the Mac and Lisa shipped was this notion that cut and paste was going to be ultimately dissatisfying. Right. Because at the end of the day, as soon as you saw something, you wanted to do something with it. I mean, the point of point-and-clicking is that something happens when you click, right? And so just-- and the other thing too is that remember, the early Mac in particular, and Lisa, to some extent, because it was lobotomized by a internal decision about multitasking, is until Andy wrote the switcher, you could only run one application at a time. So the notion of, like, moving stuff around, it just wasn't possible. And so we knew that was going to be limiting as soon as people, I mean, got over the initial fun of it.

And I think over time it became more and more obvious to more and more people because more people had experience with the interface and said, "Hey, I'd really like to do--" X or Y. And the other thing too is remember that object-oriented tools and environments began to appear. C++, all that stuff. And more people were coming out of school who understood that technology. Remember, I mean, the guys we hired learned it or invented it, in a sense. And it was only in a couple of places. So it wasn't like a lot of people were simultaneously coming up with this idea. They had to see it from other people. So I think the combination of it was the next intuitive step and it was also there was a base of programming tools and programmers who would see how they would accomplish it. And those kind of fit together. And the other thing too is that, and particularly for object-oriented development environments, the apps development stuff, Mac App, for example.

**Hsu:** Right. The frameworks.

**Pohlman:** Frameworks.

  

**Hsu:** Yeah.

**Pohlman:** Yeah. Those frameworks. Those were designed because the underlying stuff was hard. And so if you could build… I mean, it's like scripting tools. It's funny. I laugh because my son is a freshman in college, right? And from his perspective, he has no idea how many layers above the OS that he's working, right. So when he <laughs> writes in JavaScript it's like, "Well, somebody's got to interpret that and then somebody's got to do this, and then somebody's got to do memory management and garbage collection." Yeah. And all the way down to somebody's got to take an interrupt and respond to it, right? And so they just, they're, they don't know, they don't care. And they shouldn't have to. But when we were working in the early '80s, you had to care, because number one, you didn't enough processing power to do anything without being extremely clever. And second is you didn't have tools enough to abstract yourself from the OS and the hardware enough to make life easy.

**Hsu:** Yeah.

**Pohlman:** That's a long-winded answer to a short question.

**Hsu:** <laughs> No, that's great.

<off-topic conversation>

**Hsu:** This is the first time I've ever heard of the Apple IV, so I would love to know about that. And also you mentioned you worked on the Apple IIe, and I'd like to hear about that as well.

**Pohlman:** Right. Okay. Well, let's start with the Apple IV, since that's the shortest story.

**Hsu:** <laughs>

**Pohlman:** So I was, so that would've been in early '81. It became pretty obvious that the III, the III was already underpowered, you know, almost when it was launched. But the notion of expandable color slots. Remember, the Apple II was… <laughs> The Apple II and later on the Apple III were the darlings of Siggraph for years. You went to Siggraph, all the cool stuff was happening there, because it was happening at a price point that was 1/10$^{th}$ of the ordinary price point of a Siggraph classic computer, right. So I knew that there was enormous demand for that kind of thing. We also had artists who'd worked in the Apple II and had run out of gas. They needed more resolution. They needed more performance. They needed… Because they wanted to use it as a creative platform. And then we had the whole numeric and engineering and scientific community that was also running it. They had IEEE-488 boards and they had all this other stuff, but again, they were running out of gas too. So I was basically into what's beyond this platform. But still was faithful to the original Apple II vision. Versus what was clear that Steve was trying to build, which was, again, these appliances. Not a bad thing, but not what I was interested in. So the Apple IV was envisioned as a 68000-based machine. It was going to run a full multitasking, multiprogramming environment. Among other things, that's because it made programming simple. I mean, if you ever used the A-in, A-out, B-in, B-out serial interfaces on the Mac, you had to literally program byte by byte. There was no decent driver or OS to manage interrupts for you so you could just read, right? So the more multitasking you had in the machine, the less you had to worry about

  

what the hardware was like. So that was the genesis of it. I went off with another guy who's an engineering guy and we started thinking about form factors and stuff like that. So one of the other things Apple was not building at that point that was clearly going to be necessary was to put that also into a rack mount or server kind of format. Because it was a great general-purpose computer. The Apple II, for example, I did the OEM program for that. So we sold motherboards. At enormous profit. I remember going to Scotty <laughs> saying, I said, you know, "We can sell these motherboards and stuff." And he said, "What are you going to price them at?" And I said, "Well, I'm just subtracting the cost of the power supply in the case, and I'm selling it for the rest." He said, "You're kidding. You can get that much for it?" And he signed it literally in the hallway. Because they were being used in drill rigs. They were being used to target neucular missiles in Germany. It was a general-purpose computing platform. And so was the Mac, except that you couldn't get to that general-purpose computing platform. So the IV was going to be that. So we saw slots. But kind of a clever slot arrangement. So the original IV was going to be a fairly self-contained machine with an LED display that would plug into-- remember the old Apple, the Mac Duo?

**Hsu:** Oh, yeah.

**Pohlman:** So the original concept for the IV was it would plug in to a chassis, in a sense, that would have hard drives, slots, interfaces and so forth. But you could also pull it out and you could plug it into a flat-panel display or other kinds of situations. With the little things on the side, if you popped them loose you could put a rack mount clamp on it. So it was going to be a general-purpose computing module that also had all this IO expansion. We also had the notion of a brick-type power supply, because at the time Apple had to ship a different computer box for every market it went to, because the plug was in the darn box. And every time we'd decide to sell in England instead of Germany, you had to take the darn box apart. So as a product manager I said, "That's crazy." So we were basically going to ship a box and then we were going to ship the appropriate power supply. Actually plug-in in the bag. It was also going to run off 12 volts, so you could run it off a car battery, or a lot of things like that. And so that was the design. Some of that stuff wound up in the IIgs. Some of the ideas wound up in the Mac Duo. So it kind of survived in some of the concept stuff, but at the time it was designed basically to be a general-purpose color graphics machine that was infinitely upgradeable and had an operating system that could take advantage of the hardware. Again, as I said in the business plan, so that "our customers can continue to surprise us about the uses of personal computers".

**Hsu:** Right. So very much in like embedded uses, a lot of them were?

**Pohlman:** Could've. Could've been. I mean, remember, the OEM market, I mean, at the time, and this is like early '80s, the notion of having a self-contained programmable computer system that was hardware extensible, meant you were going to bury it in all kinds of unimaginable applications. Again, the original drill platforms that did slant drilling offshore, were run by Apple IIs.

**Hsu:** <laughs>

**Pohlman:** Because, I mean, what were you going to do? Bring out a mainframe out to a platform? No. You took an Apple II out to it, you plugged it in, you put some analog boards in it and stuff, and off you

  

went. So that was, the problem was, that again, it ran out of gas because ultimately people wanted more processing power than you could get in that factor. And a better programming environment.

**Brock:** Right.

**Pohlman:** You wanted Unix. Our notion though was that AT&T at the point had a monopoly on Unix and they were charging significant amounts of money. It wasn't going to work in the PC environment. So we needed a Unix clone of which there were several available. So we could break their monopoly on Unix. And put a graphic interface on it, which, obviously, was, it was ready for. <laughs> In fact, it was funny. There was a whole faction inside Apple, particularly in the mid-'80s to late-'80s. It said, "The Mac is beautiful; Unix is ugly." Well, that's because they didn't understand the difference between the C shell or the Bourne shell and the operating system, the kernel. The Kernel's not ugly; it's beautiful. You're just not ever supposed to see it. You know, what they saw that was ugly was the C shell. Well, yeah, that's just one shell of a gazillion, including, obviously, today, lots of graphic stuff.

**Hsu:** Right.

**Pohlman:** Yeah.

**Hsu:** I guess I just wanted to hear more about your involvement with the Apple IIe and that project.

**Pohlman:** Oh, yeah, yeah. So that was interesting. It was pretty clear from the Apple II standpoint, the original configuration, even the Apple II+, was memory limited and you had this kind of clumsy language card in order to even run Pascal, which made it less reliable. And the technology from the first time-- I remember the original Apple IIs, every chip was socketed, because chip failures were expected to be high, you could repair it by literally popping chip and putting new ones in. But the technology had moved on, and by '81, '82 time frame, you were starting to see the first of the large-scale integrated circuits. So jamming a bunch of chips into one custom chip. So Walt Broedner and Wendell came up with this idea of basically reducing the Apple II chip count, getting the price down, but also incorporating a bunch of functionality that was in the language card and other kinds of things to make a simpler, more powerful computer. They also had learned from the bank switching stuff that they'd done in the Apple III that there was a way to reasonably extend the memory past the 64K limit of the 6502. So all of that stuff kind of came together. And Broedner was keen to do this custom chip and so this was the ideal thing. We also needed to put a bunch of other stuff that had been kind of in software, you know, or roll it into the ROM. And produce a kind of a cheaper version of it, but at the same time keep the same look and feel. And build in the 80-column card and a bunch of other stuff that, again, from a competitive standpoint, were starting to really create problems. So this was going to be an ideal thing for the education market and for the Apple II loyal base. We cranked the project up. It's funny, because it was really Woz and a few other folks that kind of championed the notion against a lot of resistance that we should invest anything at all in an upgraded II. Again, Steve was into, you know, build a cathedral on the ruins of the previous one. So it started really as a skunkworks project, and it wasn't until Broedner had prototypes of the custom chip that it got enough buy-off. And then a guy named Dave Larson, who wound up with quite a career at Silicon Graphics, who was a great project manager, he seized the reigns and did a perfect launch of the product. And the timing was good too, because Lisa was coming out, there was a big window of attention. And it

  

told the marketplace that the company was still committed to the technology. So yeah, it was fun. I, other than getting it off the ground, and kind of shepherding it along a little bit, the actual details were left to others, but it was the right thing to do. We had created a plan for the II, and basically between the IIe and the portable II, the IIc.

**Hsu:** Was that the IIc?

**Pohlman:** The IIc. Those were the two evolutions of the II that we foresaw back in 1980 when I took it over, and both of them saw the light of day.

**Brock:** Wow. But I did want to, if I could, just ask you maybe some concluding questions.

**Pohlman:** Sure.

**Brock:** These are just kind of big <laughs> think piece sort of questions. And one that's just looking back on your engagement, actively today looking back to the start of your engagement with software and computing into the 1960s, you know, just to what are some of the things that have impressed you most about the changes over that time period of, in the nature of software, computing, its place in society, our world? That would be to reflect on that, and then thoughts for if there were a younger person listening to this, watching this, thinking about--

**Pohlman:** Most of them will be. Right?

<laughter>

**Brock:** Technology, you know, as a possibility for them, as a career.

**Pohlman:** Oh, sure.

**Brock:** Or an area of activity for them. Any thoughts or advice you might have for somebody like that listening.

**Pohlman:** Sure.

**Brock:** So…

**Pohlman:** Well, the evolution of software has really been all about layers. So when I think about that, and when I describe the tape-based operating system and compiler where you could actually see a compile happening, right, to the point now where compilation is, like, a mystery nobody has to even worry about anymore.

So the amount of computer power available means that the abstraction of application tools is almost ultimate. So at the end of the end of the day, for better or worse, people don't have to know how computers work anymore. And this whole notion of MIPS per milliwatt, right, which we were so worried about. You know, how do we squeeze the maximum amount of power before the battery goes dead? Those become irrelevances, which is powerful. Now, in a sense, it would be great if every kid who used a

  

computer or a, you know, an iPhone or whatever, actually knew how all those layers worked, right? What's an interrupt? <laughs>

But the power is that they don't, which has unleashed millions of programmers as opposed to hundreds or thousands. I mean, when I was starting out, you probably couldn't count all the programmers in the world, you know, very effectively, but you could come close. When I was writing articles on Business Basic in Softalk magazine, a lot of the stuff I was doing there was borrowed from Knuth. And who remembers him anymore, right? I mean, the three-volume set of algorithms and stuff like that. You know, how's sorting done? What's a B-tree? You know, those are all so buried now that we're just focused on the work. I mean, you can go to an app development class and use an app development tool on an iPhone and make an application. You really don't have to know much about the computer other than what is it you want to do?

So, I mean, the ultimate computer interface, remember, is you just lean your head against the screen and say "do what I want", right? But we're getting awfully close. Awfully close. And I think that's embedding software into our lives as opposed to seeing it from a distance or understanding that it's this kind of different creature. You know, the bits and bytes and stuff like that. It causes me to be astonished that today's computers work at all. On the Mac there used to be this little application called Googly Eyes, and it sat on the <laughs> top line and kind of winked at you and whatever and rolled its eyes. And I never ran that stuff, because I knew how the OS worked. And I thought, "Man, that is just asking for trouble."

**Brock:** <laughs>

**Pohlman:** But today we throw stuff in and the systems just adapt. So I think there's enormous power. I think if I were a kid today, hopefully there would be some natural curiosity to kind of understand that or maybe read some of the literature. But on the other hand I'd say, "Hey, don't even worry about it. Just go make stuff work that you want to do. Bring creativity about what it is you want to accomplish as opposed to how it is you have to accomplish it." So much time was spent in my day on how you get something done and much less about what it is you want to get done, right?

And so when I talked about, for example, we had to create Foundation using a C compiler because we didn't have C++, which meant people had to do object-oriented programming in their brains and translate it. Way too hard. No wonder we failed. So let that liberation, just kind of enjoy it. It would be great if you kind of appreciated <laughs> it a little more, and how hard this stuff is at its gut. But on the other hand, the reliability of those lower layers is now so high that there, that debugging in the classic sense, the blue screen of death or whatever, is so uncommon now, that we just don't have to worry about it so much anymore. So to me that's incredibly liberating.

END OF INTERVIEW

  