PC Software Workshop:
Financial & Management Software

Moderator:
Luanne Johnson

Recorded: November 19, 2004
Mountain View, California

CHM Reference number: X4408.2008

Table of Contents

PC Software: The First Decade – Financial Management Software Workshop

Conducted by Software History Center – Oral History Project

Abstract: A group of experienced pioneers in PC software discussed various aspects of producing PC financial and management software. They talked about financing their companies and the initial products which they developed and marketed. They reviewed how DEC had misread the market opportunity with their Rainbow PC and some of the other failures because of not matching the application functions to the capabilities of the machines available at particular points in time. Other topics were the choice of languages used to implement the programs, spreadsheet competitors and expanding the businesses internationally. There were comments about multiple application suites and the discussion closed on the wisdom of hiring former IBMers to work in PC businesses.

Participants:

Name	**Affiliation**
Luanne Johnson	Charles Babbage Foundation, moderator
Robert Carr	various
Karol Hines	Ross Systems
Gib Hoxie	Contact Management Systems
Mitch Kapor	Lotus Development
Lee Keet	tsi, NCSS, D&B, Vanguard Atlantic
Jan Phillips	DEC
Ken Ross	Ross Systems
John Toole	Computer History Museum
Paul Ceruzzi	Smithsonian, historian
Jim Cortada	Charles Babbage Foundation, historian
Thomas Haigh	University of Wisconsin, historian

Luanne Johnson: Before we start, I'd like everyone to introduce himself or herself, let us know what your affiliation is and so on. I'm Luanne Johnson and I'm with the Charles Babbage Foundation, and I am the Principal Investigator on the IT Corporate Histories Project, which you will hear about at lunchtime. And I was also, with Burt Grad, the co-founder of the Software History Center which is now, I'm very pleased to say, part of the Computer History Museum.

Paul Ceruzzi: I'm a curator at the Smithsonian National Air and Space Museum in Washington. And I guess I'm best known in these circles as the author of a book, *The History of Modern Computing*.

Ken Ross: Timesharing, minicomputer and microcomputer software, currently in venture capital.

Karol Hines: I was formerly of Ross Systems and several other things. And currently I'm working with Luanne on the IT Corporate Histories Project.

Jim Cortada: Chairman of the Charles Babbage Foundation and more importantly, I think, a historian.

John Toole: I'm Executive Director of the Computer History Museum.

Thomas Haigh: I'm currently with the University of Wisconsin, Milwaukee, and the Haigh Group, a little historical consulting thing.

Jan Philips: I'm a retiree of Digital Equipment and Compaq Computer and HP, and who knows what else…. <laughter>

Robert Carr: Back in the early 1980s, I worked on a couple of different PC software programs, one of which Gib Hoxie will mention. It was a competitor to Mitch Kapor's Lotus 1-2-3. And then I went off on my own and developed a product that ended up being a strong competitor to Mitch's Symphony. So it's appropriate we're across the table. <laughter>

Gib Hoxie: With Brian Fischer, I founded Contact Management Systems and Context MBA. And we were promptly overwhelmed by Lotus 1-2-3 and rightly so I might add. Not any smarter, I wanted to do a Macintosh software product which was purchased by Doug Bergen, and marketed as Plain & Simple. A lot wiser this time, I went back to consulting. I'm now retired except for my participation in Tech Coast Angels, an investment group in Southern California.

Johnson: The management of this conference gave us all sorts of guidelines as to the kind of questions to pursue. I think that they're important questions, and I think we should address some of them. But I also believe that there will be other questions that will come up as we go along, particularly, from any of our historians who really feel that there is some particular area that they want to hear covered. But, just to get things rolling, I'd like to go to this list of questions, they're in your packet, and start with the first one. And then we'll see where it goes from there.

It's kind of interesting, because the first question is “How were these early PC companies financed?” And since we have such a variety of examples here such as Ross Systems, which was a well established firm in time sharing, and then in minicomputer software, and then went into PC software. And then we have the DMV experience where we had the use of a big company that was branching out into PC software. And we also have examples of where the PC software was the start-up kind of environment coming from some place else. So let's talk about how did this whole thing get going and what was the financing behind it and so on. Mitch, would you want to kick that off since you're at this end?

Financing Lotus Development

Mitch Kapor: Sure, can I ask how long an answer do you want? <laughter> No, I'm serious; it's like an unfolding set of Chinese boxes. What about the two minute or less answer?

Johnson: Well, let's start with the two-minute or less answer.

Kapor: How are these companies financed? The first thing to say is that the financing of Lotus Development Corporation is the subject of one of the best known Harvard Business School cases that was written at least 15 years ago and is still being taught at all of the major business schools and so it's amply documented. And the short answer is that Lotus itself was a classic venture-backed start up, financed by Ben Rosen of Sevin Rosen, with co-investment from me. The annotation I might make to that is that I had done a number of things prior to Lotus in fairly quick succession, one of which resulted in my having some money, because I sold the rights to VisiPlot and VisiTrend back to the publisher, Personal Software (later VisiCorp), which left me with a bunch of money. And I took $300,000 of my own money, which was a lot because I had never made more than about $14,000 a year, to be the seed money for what turned into Lotus. Except I didn't know it was going to be Lotus, because we didn't know what the product was going to be. And I went out and I really burned through most of the money before we had much of an idea of what the product was going to be. I was terribly scared of venture capitalists, having run into Arthur Rock in the six months I'd lived in California, in 1980, and worked as a product manager for Personal Software. But I did know Ben Rosen, and he seemed like a nice guy. In a very naive kind of way, I sent him this long letter, which is attached to this Harvard case, that starts out in a famous kind of way saying. “I just want you to know that there are some things that are more important to me than making money.” So let's get this straight. This is me pitching the venture capitalists. <laughter> So, it's in the case, and then I decided to take a chance because we really believed in the PC software category, and I had actually done a couple of products that had been successful.

Johnson: What year was this?

Kapor: 1982 was the year that I pitched Rosen, early in the year. And then I found out he wasn't any more experienced as a VC than I was as an entrepreneur. But, he had confidence that I and the other key person, Jonathan Sachs, who was the technical architect, could do the right thing, and he put the money in.

Creating Duns Plus at D&B

Johnson: Well, I'm sure there's more. Lee, tell me about two of your experiences: the one with Dun & Bradstreet, and then when you started the PC software company.

Lee Keet: Well, just one sentence. I have had four careers; the first one was with IBM and the second one was my own mainframe software company [tsi] which was purchased first by national CSS and then by Dun & Bradstreet, and they asked me to run what became known as the Software Products Group. And I ended up with three divisions, the last of which we acquired, which was McCormack and Dodge, (In the Annals it's wrong. It says that was 1985; it was actually 1982). But I was playing with VisiCalc on my own, and had this idea that you could build a complete tax preparation product in VisiCalc. And I actually did a prototype of that and this was on an Apple III. It was my first exposure to PCs and PC technology, and I went to D&B and said, “I think that there is an opportunity here to build a platform, a hardware/software platform, looking back you would call a suite, where you would have word processor, and a spreadsheet, and all the glue around it.” And we started an internal start-up called Duns Plus.

We then went and did a deal with Don Estridge, who was running IBM's PC division, and I said, "Don, I want to brand this thing. I don't want the IBM label on it. I want the IBM label and the Dun & Bradstreet label." And he says, "IBM doesn't do that." And I said, "Well, why don't you go find out what it would take." And he came back and he said, "IBM would like earnest money of a million dollars, for which you get nothing, and you can do it." So I went to Duke Drake, and said, "Duke, I need a million dollars to do this," and Duke had a check for me within an hour. And to make a long story short, we actually built the Duns Plus product. It was launched after I left D&B in 1983. But the original platform was going to include some of the leading software of the day. VisiCalc was the one we had chosen, and I was negotiating hot and heavy with Dan Flystra for, not only the product, but we actually got seriously negotiating about buying the company. So, you can see here, this was a deep pocket organization that was going to try to make a success out of it. At the same time, I got permission from D&B to take my little tax calculation preparation product and start a company to do it, which I did myself. And that was my first exposure to venturing.

This tax preparation software was called Soft Tax. Eventually, that company was purchased by an accounting firm in Texas. It was early to market. We had an actual application product in a slip sleeve by the end of 1982. The finish with Duns Plus, just about when we were about to pull the trigger on VisiCalc, Lotus 1-2-3 was just getting going. And we got an early copy to play with and said, "This is dynamite; it's a much better product." Mitch came to me and said, "If you do another project, come to me for funding okay?" But Lotus was really easy to do business with because I think -- I do not remember this exactly -- I think we agreed to buy thousands of copies of 1-2-3. But we got it for five dollars a copy.

Ross Systems Enters the PC Software Business

Johnson: Ken? Tell how you migrated to the PC business.

Ross: We financed our company mostly the old fashioned way, through profits. We started from scratch as a consulting company, got into the timesharing business, which today they call “on demand” and they think it's a whole new computing metaphor. But this was 1975 and it was the timesharing business. You could log on every day and see how much money you made. We were profitable and the business was going great up until the late 1970s, early 1980s when these things called microcomputers started showing up. And I remember that we could bill some local big companies -- Intel was a big customer, I don't remember how much they billed, $40 or $50 thousand a month for everybody logged on doing stuff. But we were able to use the profits to finance the development of our microcomputer software product which ran on the DEC Professional 350, so that was sort of the claim to fame. I have the video here, which we can look at in a little while if you want to.

Johnson: What is the video?

Ross: It's actually a Ross product advertisement, five minutes, that was an ad that we made but it sort of teaches the use of the product.

Johnson: What year was that?

Ross: This was I think around 1983. But it's sort of interesting for me to hear the talk about the DEC Pro 350, and the features, and look at the relatively rudimentary kinds of things. Yet when you hear the kinds of things that it does, consolidation and stuff, it's actually still sort of pretty relevant. Just the technology's old. We actually raised a first round of venture capital sometime in 1982 or 1983 just because we were expanding; we were trying to do multiple things. And that seemed to be the way to go. There were not very many venture capital firms and that seemed to be the time to do it. And Bruce Anderson [of Welsh, Carson, Anderson] was the guy who made the first investment in Ross Systems; they were a big, big, early IT venture capitalist. But profits were the way to go.

Johnson: Does anyone have anything to add to that?

Hines: The only thing I'll add is that the product that we put on the DEC Pro 350 was sort of a spreadsheet that we had developed and where we made a lot of our timesharing money was through maps.

Ross: Yes, it was a port of our time sharing software.

Johnson: Okay. So, with the timesharing services you were selling this application. It was like a financial modeling package?

Ross: It was financial modeling and consolidation, and it moved from timesharing to microcomputer, and then we had another version that ran on the DEC VAX, and so the whole theme of the Ross Maps product was more mix and match between minicomputers and microcomputers. It's kind of interesting to see since it goes back a while. I'd say 1983 was about when we were doing it.

Hines: And then that product also got migrated to an IBM PC later on in another life.

Ross: Well, it was a bit different, yes. I stayed involved with budgeting software, a different product, but finally I got out of that. I got bored.

Johnson: Robert?

Starting Framework

Carr: Yes. Well, I see I've probably got a different kind of financing story. Mine is kind of a starving artist who hits it big through hindsight, and probably a lot of luck. In early 1982, I had the idea to build a product that became Framework, an integrated package that included a spreadsheet, word processing, database, and graphics. This was kind of the Holy Grail back then. Because PCs only had one or two floppy drives, and you didn't want to have to quit out of 1-2-3 and then run WordStar; it was kind of a pain. By summer or late 1982, I stopped consulting. I had been consulting for Gib's Company, and I moved to San Francisco, rented a kind of a "starving artist" apartment, and started living off savings to program this thing. And by not knowing exactly what I was going to do, but knowing that I needed to, because the whole vision was that it would be a very well crafted, tight program. So, I wanted to work by myself and get it built that way. By the summer of 1983, I had what I thought was almost finished -- in hindsight, it was really just an advanced prototype – and I started showing it to friends.

One friend said, "Do you know what you have? This is really something. I want to join up with you." So I got a co-founder out of that, it was actually somebody else who had been working for Gib who had business sense. I was the naive programmer.

Johnson: What was his name?

Carr: Marty Mazner. I was the naive programmer who had just lived off my savings. He had the business sense, and he said, "You know, Robert, I don't really want to go to IT venture capitalists, because they'll take our hide. And also, the industry's maturing so that it's probably too late to try and be our own software publisher." This is, remember, the summer of 1983. He said, "Let's have a different strategy. Let's go to one of the top software companies and get them to publish our product but fund us as a separate development company so we can still have upside potential." So we looked at all the companies and decided there were two that we thought would be the best partners: Ashton-Tate, and Lotus. Mitch, I don't know if you remember this, but Marty called Mitch up, and said, "We want to show you something. We think this could be a good follow-on product to 1-2-3. We had a date to meet Mitch about a month later when he was going to be on the west coast. And Marty then called up Ashton-Tate. They said they had no time whatsoever to talk to us, but he insisted, insisted, insisted, so it was kind of, "Okay, you get one hour." We flew down there and I demoed the product to them, and it was one of these things where they kept bringing in more executives every 15 minutes, and they said, "You've got to stay in town. You can't leave." And by 24 hours later, we had a deal with them where they financed our company to finish the product, but allowed us to stay as a separate development company where they got the publishing rights. So it was a different model than venture capital financing. And it ended up working well for us because we got to grow our company for two years while they were the publisher. They made the product very successful and then two years later they bought our company, once it was proven on the market.

DEC's Failure in the Marketplace

Kapor: In working with DEC there was a guy who was the general manager for the Rainbow Division who was my liaison. And I had already had a number of meetings saying, "Your guy's strategy makes no sense. You've got these three different incompatible machines. You've got the 350, you've got a retooled word processing machine, and you've got the Rainbow, on which the featured OS was CP/M. What are you guys doing? You're blowing a big opportunity." So we didn't resolve that. I got someone out to see Ken Olson in his office, and he spent the first 15 minutes of the meeting on a long discourse about the relative quality of the cases used on personal computers. He had an IBM PC case and a case for one of the DEC ones, which he had designed personally. And he was talking about the thickness of the sheet metal and that the IBM PC was flimsy. And I couldn't tell what was going on. I thought, maybe it was like reality TV or something, Candid Camera. <laughter> No, he was into that computers are industrial strength, not these crappy plastic consumer products, and it was clear he was so wrong, you couldn't even start talking to him about. His attitude was: PCs are toys; they'll never amount to anything. That's a real computer? And I just thought, if I'd had any Digital stock, I would have sold it that day, because it was clear the captain of the ship was saying, "I see an iceberg, and I'm heading right for it." I mean, he was off in every imaginable way about where the market was, what the opportunity was. People bought personal computers, and that was what was going to make them successful. And all of a sudden I understood why they had a completely stupid strategy.

Hines: He took great pride in the fact that he didn't have a computer in his office.

Ross: Did he ever know, or did they ever realize that MS/DOS was really based on the RSX11 operating system?

Kapor: Oh, sure, people in the organization figured that out.

Ross: And Microsoft Basic had a lot of characteristics from the PDP-11 Basic that was originally done, too.

Kapor: Yes, so it was. I'll also say that in the same timeframe I met Dr. Wang. And he was more soft spoken, but not any smarter about what was going to happen in the computer business. And, you know, Boston was the hub of all the minicomputer companies.

IBM and Xerox in the PC Marketplace

Ross: But, when the big computer companies became successful they became arrogant, and it happened to IBM, too, at the end of the 1980s. You know, they think they knew everything. And the executives, I remember reading that − was it Akers that was the CEO at IBM? − neither he didn't nor any of their executives had PCs on their desks. This was in the early 1990s when everybody did. So they had lost touch with reality.

Carr: Maybe in terms of technology roots, certainly I have to give credit to one that was key for me − Xerox − which is a company in contrast to DEC, not that they were successful in the end. And maybe this will stimulate some other comments. When I graduated from Stanford with -- back then they didn't have an undergraduate degree in computer science, but I kind of forced my way into it in engineering. And when I graduated, in the last few weeks, rather than taking a mainframe job with a company on the east coast, my professor introduced me to some folks from Xerox PARC who were creating a new kind of spin-off from Xerox PARC to try and commercialize their technology. So, I took that job right here in Palo Alto, and for the next year I worked on the famous Alto computer, which, back then, was everything computers still are today, except for one thing: they didn't have spreadsheets. But we had a suite of office software that was well integrated, the graphic interface, the mouse, the Ethernet, the ARPANET, email, laser printers – it had everything. It was just amazing. So, I worked at PARC for a year in 1978.

And back then, that vision had already been completely realized for at least a year. They had these systems working in their pure research center. And in early 1978, they recognized that they were having problems commercializing this technology to take advantage of it. And one of their attempts was to create an intermediate organization that would start with their quick prototype and take the technology out to market in limited quantities so they could learn while their commercial division in L.A. was building what ultimately became the Star system, that ended up being too big and heavy to ever be successful. So they brought together a hundred of their best people from PARC, put them in this outside building, hired some young programmers. So here I was, this little fish in a big pond.

But almost the same year that Apple was founded, there was this fully instantiated vision of what personal computers should be. And that was certainly in the collective unconscious, certainly out in the Valley, and many places, and it was very specifically in some of our heads. So that in my case, four years later when I was coding Framework in assembly language on what was actually quite a crude little IBM PC compared to this Alto, I still had this vision of where we needed to head. And I did my best to realize that, on the crude machine, doing things like implementing pull down menus in Windows even though the IBM PC didn't have a graphics system.

Ross: It always takes ten years for the vision. I mean, I always think of the year of the LAN, which I think occurred in the early 1980s. And it took ten years.

Kapor: It was annual event, actually. <laughter>

Hoxie: There's a corporate cultural issue that Robert is heading on. When Xerox was trying to sell these GUI machines, they were selling them to corporate managers, who were buying them for secretaries to use. So, the lowest-priced people in the office were being asked to use this $100,000 tool. Not going to happen. There's no amount of productivity gain that could pay for that. And so they weren't being bought. Executives and professionals didn't have the spreadsheet tool at that time that they needed in order to motivate them to end run MIS and the corporate buyers. And through the back door Apples came in, which everybody started using with VisiCalc. People were reaching into their own pockets, buying Apple IIs and VisiCalc and running them into corporations and hiding them. Because the MIS guys were going in with sledge hammers trying to kill them. Xerox didn't have the $2,000 tool that really solved the managers' problems, to have the economic power and reason to buy it. And the secretaries certainly weren't going to benefit from these expensive Xerox systems.

Keet: So, the word processing was not a big deal because you had fairly low-paid people who would type your letters for you.

Hoxie: Yes, there was Wang and there were a zillion cheap computers. So they were way ahead of the market need and ready.

Kapor: That's why word processing wasn't a breakthrough application.

The Need for More PC Power and Memory

Ross: I remember a Comdex when VisiCorp was showing their integrated VisiOn on an IBM PC that had zero power, so that you couldn't do it. But I remember Fylstra trying to demonstrate that.

Kapor: It was limited by the power of an IBM PC. There was a mismatch between the way VisiOn was architected and implemented and what the machine could do.

Carr: They architected it the correct way, but they would have needed to wait another five years. Mitch and I were doing the bare bones, bare model.

Kapor: There were different traditions then about what the correct way is in terms of good software engineering practice. Things like VisiOn were done more correctly. They took people out of Intel and other places that had been doing development. But it was another tradition, which is a minicomputer tradition, where the key software people had started trying to cram the equivalent of mainframe operating systems into very, very scarce resources. So, Jon Sachs was used to working in environments with 64-kilobytes, 128 kilobytes, because that's what he had to implement a piece of system software with when he worked for Data General. And that turned out to be the right approach. It was impossible to communicate with them because they were so sure about this. Oh, you kids, move over now, we've got our venture capital. We've got our professional management team. We know what we're doing. We're going bring in the people who really know how to develop software. Those are the ones who did VisiOn, and in fact, one of the things that encouraged me, to make me think that there was some possibility that Lotus could succeed, was seeing the blatantly self-destructive approach that the principal companies in the field were then taking. It was, well, all right, if you're going to commit suicide, fine.

Cortada: Were you still at VisiCorp, when VisiOn was done?

Kapor: No. I wasn't. We were doing Works. I think Lotus came out before VisiOn. They were years later.

Keet: I think these two dynamics are important for the historians, the Moore’s Law dynamic. And in those days, you were working with such a little bit of power. I think it's like

64,000 times the power increase from the last 20 years, and so little real estate to put your software in. So, you put that into context to what these companies were doing, it was a major issue; and the other one was computer literacy. I remember taking Duns Plus. We figured if all the executives at Dun & Bradstreet weren't using it we're never going to sell it. So I ended up personally trying to train Charlie Marks who became the Chairman, and Duke Drake, who was the Chairman, and all those guys. And they had never touched a keyboard or a computer in their lives, and in the end, they didn't want to. They wanted their secretaries, their administrative assistants, to use them. And if you put that in the context of what people were trying to do then versus what they're trying to do today when an SR1 for Office XP IS 22 megabytes. And people are all computer literate now; it was a whole different world.

Languages for PC Programming

Hoxie: On technologies, I want to add one thing to the notion of choice of platform for compatibility across a question of computers. There's one other issue behind our choice of Pascal as our language. It's something called data typing. Unfortunately, this term has fallen into disuse. But it is the reason computers crash, because we use languages that allow people to say, oh, the seventh bit of that particular data item can be used to change the date. Okay. And you go stick that bit with something, you flip it, and suddenly, the data no longer mean what they used to mean or should mean, and so systems crash. Well, because somebody else wasn't prepared for data.

Ross: That was a whole Y2K issue.

Kapor: Whoa, whoa, whoa. Oh, this'll be fun. <laughter>

Hoxie: So, Pascal had strong data typing. Okay? You could not control bits in there without really standing on your head. And so we knew if this data was a date, it would be a date. Okay? And so we felt we could have a robust -- actually, debugging our software went very smoothly because we eliminated all those data conflicts.

Kapor: This debate about developing in weakly typed or non-typed, and strongly typed languages is still going on today. It hasn't changed 20 years later, and today it's framed in terms of Java versus Python. And we are once again, as we were back then, on the opposite side of it.

Keet: Except now you have infinite memory and huge storage.

Kapor: It doesn't matter. I'll tell you why.

Keet: But Java wouldn't work in the old days because it works...

Kapor: No. That's not what the debate is about. It's about programmer productivity. Our guys can write ten times faster in Python than they can in Java. The overhead, to set up the management, what you have to do, especially as you evolve and refactor your software and rewrite as you go along, you effectively can't do if you are writing in Java, whereas in a scripting language like Python, which lets you do programming in a large area but does not enforce typing -- and this isn't me saying but these are some of the world's best programmers saying, I love writing in Python now. It's not like there's a right answer to this, that the strongly-typed languages have advantages that are good for certain things and the untyped ones have a different set of advantages and constraints. And it's not an issue that is just about 1982. That's my point. It's a perennial issue that keeps showing up.

Ross: Like the year 2000. A lot of what happened there. That people decided that they were going to be more efficient or faster.

Hoxie: How about viruses and worms. I mean, I think it's just a part of the whole picture

Kapor: Yeah, well there's a place for [strong data typing] -- I think the one thing I will say is that if you're going to do system level software or where you're concerned with just extraordinary robustness, there are strong arguments to be made for writing stronger type language. If you're trying to write application software for individual productivity users, there's an argument that says trying to write something that maximizes rapid development and programmer productivity has huge advantages. So, I would be interested in knowing what Robert thinks about that.

Carr: Yes, yes. <laughter> There's a place for both and I think it's a healthy tension. It won't go away 20 years from now, either.

Kapor: No, probably not.

Spreadsheet Competitors

Haigh: So, I've got another question. And there were obviously a lot more spreadsheets on the market in the early 1980s than just Lotus and VisiCalc. And I don't know if there were more integrated packages than just simply in Framework. So, can you mention any of the competitors that you think were significant and might have been viable competition if they'd done one or two things differently?

Kapor: MultiMate. Was it MultiMate?

Ross: MultiMate was a word processor.

Carr: MultiPlan, which is the one I left.

Kapor: Which was from Microsoft.

Carr: I'd like to get your thoughts on why Microsoft's spreadsheet lost out to you for so many years.

Kapor: Well, if when we were designing 1-2-3, there were certain principal products in the market: Context MBA was in the market; VisiCalc was the leader; Microsoft had its product. But again, there wasn't a kind of a consensus about what feature set and behaviors really mattered. People had different ideas about that. And so some of the products that were out early like Multiplan just kind of doomed themselves from the start by being really techy. VisiCalc invented, to my knowledge, the cell reference, A1, B5, using a letter and a number for the row and column. Okay. With Multiplan, the fundamental approach was involved with a bracket, a number, a comma, or some sort of separator, and another number.

Carr: R1, C1, row and column, you had to put square brackets around them. That you didn't need dollar signs I think was their key point

Kapor: It was baroque and complex and hard to grasp, and it was just wrong.

Carr: But it was more efficient, it was like in the early 1970s, when HP came out with their calculators that used reverse Polish notation. Then if you could learn it, it was much more efficient.

Kapor: But the market of people who were going to be the early adopters, that was a bad match for them. A1 and B5 just worked infinitely better for the early adopters because they were not computer professionals. They were people who were doing spreadsheets for a living. So, it was kind of like Stephen J. Gould wrote in his this book, I forget the title of it. But he wrote about the great explosion of species at some point, there was a Cambrian explosion where they were many mutations-- and most of them just turned out to be unfit or less fit for survival than the others. And they quickly were sort of abandoned experiments.

Hoxie: For historical reference, I guess that many of you have probably never seen a real spreadsheet. This is what a real spreadsheet looks like. For Booz Allen, you go to the Booz Allen document library thinking you could check out pads of this. You had to sign your name to get these sheets of paper. And these were big enough; they had enough rows and columns that you actually could do a serious financial model. They're very durable, they can be erased 50 times before the paper falls apart, because that's what we had to do back then, and you did it all in pencil.

Carr: Recalc with an eraser.

Hoxie: That's correct. That's what VisiCalc replaced.

Haigh: But the thing you've actually drawn on there is a flow chart.

Hoxie: This is an original design document from a project that I worked on

Carr: But there's a grid of columns and rows.

Kapor: Well, I will say that from the early to late 1970s, people used these financial modeling languages on timesharing systems to do spreadsheets. And good or bad or whatever, that was the norm.

Carr: So let's name just a few other competitors; let's get them into historical record.

Kapor: There was SuperCalc, which was out before 1-2-3, which was CP/M based.

Carr: Right, and if I remember, they were reasonably fast and powerful, so they got that right, but they were on the wrong operating system.

Kapor: Wrong platform.

Keet: It went fast. It was fast, and it was functionally essentially identical to VisiCalc.

Kapor: They were not on PC/DOS in a big kind of way.

Carr: And then Personal Software eventually came out with VisiOn with a spreadsheet in there.

Kapor: VisiCorp.

Carr: VisicCorp, yes, thank you. But that died because VisiOn was a giant pig that was years too late, that just was still not running fast enough.

Hoxie: Fred Gibbons did the graphics thing that would take VisiCalc data and do better charts than 1-2-3. But I can't remember what the thing was called, do you remember? Was that Harvard Graphics?

Kapor: I don't know.

Multiple Application Suites

Keet: Well, what was the 2 and the 3 of Lotus 1-2-3?

Kapor: Well, there's spreadsheet and graphing and a kind of database. There was supposed to be a word processor in it, but the guy we hired didn't work out.

Johnson: You didn't do 1-2, and a half? <laughing>

Kapor: Well, I was supposed to do the thing instead. And it wasn't really a database, but there was some data access functions in it.

Carr: Yes, you could sort.

Kapor: We had the range justify command. We wrapped a set of rows of text into a paragraph so people could use it as a rudimentary word processor.

Hoxie: But you also called it a worksheet.

Kapor: The 2 and 3 were graphing and data.

Carr: So again, back then because IBM didn't come out with a hard disk version of the PC for at least a year or two…

Kapor: There was the XT, and that came out when?

Carr: But until that arrives you typically had a floppy drive, and you booted up one application so you ran WordStar. And then when you wanted it to do numbers, you'd have to save your file, quit that program, take your floppy out and put in your spreadsheet floppy. The goal was to do 2 or 3 other products on one floppy so you never had to shut down and switch between them.

Hoxie: You want to see something really bizarre, you ought to see this thing running on a four diskette Apple to complete what it first did.

Kapor: Four disk drive?

Hoxie: Four diskettes. Four diskettes on an Apple II. <making a chugging sound> Oh, a new cell calculated. <laughter>

Matching the Applications to the Machines

Carr: I guess what we're hinting at here, what we've been touching on all morning, is really, just how much the history of software is driven by the hardware, and what the hardware enables and doesn't enable. Whether or not there's a hard drive and you have two or four floppies; boy, the software starts looking weird.

Kapor: If you're going to do a piece of software, having a good match between the hardware capabilities that you're targeting and what your software needs is critical. I mean, it's one of the worst situations; you're sort of dead in the water if it's not a good match.

Carr: But it seems that this is less true these days, though. It seems like the hardware has run ahead of the software for so long in terms of excess power, I'm not sure there are so many examples of the software really being there.

Hoxie: No, what the driver now is the bandwidth into the home.

Carr: Yes, certainly for the last ten years it has seems that it has been on the cusp of being reversed.

Kapor: It's also that you can use so many things today by gluing together different things, that you just give up so much inefficiency that you start running at unacceptable performance. And you have to take time and optimize.

Hoxie: Adobe understands how to use all the memory and computing power you can toss at it. I have one gigabyte of RAM computer to do my photo software.

Ross: I'm always amazed about the capacity of these devices. And you know, you can develop applications on these things with Linux that have five, ten million lines of code in it, so that when you develop a thing like an embedded application, you don't need to worry about the constraints the way you used to.

Keet: I'm fascinated by the trend going backwards, though, because there's now new Linux functions for example because people are trying to get bootable devices like the USB FOB so you can have a portable system that you just go from machine to machine, so we're back to trying to work in little spaces because of portability.

Carr: True microcomputers.

Keet: And I wasn't exaggerating when I said SR1 for Office XP is 22 megabytes. I'll bet all of the software that all of us developed in the early 1980s didn't get to a quarter of that.

Haigh: So how long did it take for 1-2-3 clones to come out? I think there was Easy As, and Twintopia.

Kapor: Oh, I don't know, 18 months probably.

Carr: There was this one that Adam Osborne came out with, he had the clone that was $99 or something.

Kapor: There was Mosaic, that was not made by him.

Carr: Paperback Software.

Kapor: Right, right.

Carr: So why did clones fail against 1-2-3? Certainly we've talked about distribution and marketing were things you did right from the beginning. And those would have been significant obstacles to the clone companies.

Kapor: Well Lotus, under CEO Jim Manzi, pursued lawsuits against them which had a chilling effect on those even though the ultimate outcome went as it should in the opposite direction to sort of not uphold copyright on look and feel. It stalled people for a couple of years. And that was a critical period of time.

Carr: I had a copy of Paperback Software from that time, which was in a Ziplock bag. That was like in 1971. <laughter> But I bought it in a book store or something like that. And it said everything but the price. That was what the thing said.

Johnson: I need to give you guys a break. But I'm really curious about -- now, this video that you have, did you guys make like a commercial?

Ross: This was a five-minute Max Pro/DEC 350 commercial.

Johnson: I mean was it a demo? Or was it a commercial?

Ross: No. No. It was just a videotape that we sent out to users.

Carr: Were there video machines back then?

Ross: But we had it. You could see it on VHS, whatever the heck that is.

Hines: And the salesmen had it. It could run on their machines.

Carr: That's right, yeah, we had VCRs back then.

Johnson: Interesting sales approach. So you were sending these VHS tapes to customers, is that what you were doing?

Ross: Yeah. Again, we never took the retail distribution route. We always were working through enterprise sales, working with the DEC sales force, and quite frankly, we'd rather people bought our VAX software or timeshare. But this was something we felt obligated to do because the world was moving towards microcomputers. So we always had one foot in the microcomputer software world, and one foot someplace else.

Hines: Right, some place else, maybe two other places.

Ross: We had a lot of feet moving.

Johnson: How successful was this as a sales technique, sending out this video?

Ross: The whole product was never very successful.

Philips: Apparently it wasn't so successful.

Hines: And how many Pros did DEC sell?

Ross: I have no idea how many DEC sold, but luckily, we were either smart or lucky enough to have a parallel development process going for Cobol-based ERP software for the VAX. So that adds up to being very successful.

International Activities

Cortada: Let me ask a different question. I'd like to move us to a different paradigm, since we've talked about primarily the American line. What experience did you have with your product and your services in Europe? Did you ignore it? Was it different?

Carr: My experience back in the early and mid 1980s was there was little viable competition coming out of Europe. The market over there was always a couple of years behind the U.S. market but it still was a sizable export market. So in the 1980s, the smart PC software companies quickly developed their European marketing and distribution channels, and reaped quite a bit in the way of revenues from there. The key limiter was competency. You know, if you could get over there and get good subsidiaries or distribution partners going quickly, you could get a lot of market share.

Hoxie: For us all we had to do was localize for currency, because in the early 1980s, anybody in Europe who was computer literate would do it in English. We didn't have to translate it into French or German. And there was no Asian market that we could discern, because it was just culturally completely wrong; anybody who was a manager or professional wouldn't put their fingers on a keyboard if their life depended on it. That's changed, though, but certainly not in the 1980's.

Kapor: So we made a big investment from 1983 on, when we came out in international operations. We set up companies, subsidiaries, in England, France, Germany, we started localizing the software, and we put a software development center in England to do localization work. We rearchitected the product to make that easier. We set up a manufacturing facility, given the tax incentives. In Ireland it was manufacturing in those days, with lots of disk duplication, and assembly of the stuff. And we cut local distribution deals, and it became a sizable proportion of the business, 20% or 30%. And with renewals, it had to be 50%. We also, I think in 1985 or 1984, started very heavily investing in Japan, hired a Japanese national to form a subsidiary, spent a year really redoing the product for the Japanese market. And interestingly 1-2-3J in Japan was the single best selling piece of software in the Japanese market for a number of years. It actually outsold even the word processing packages, which was the reverse of U.S. and Europe, where the highest volume in productivity tools was from word processing. They had more problems with word processing because of the alphabet issues. But the lesson for us was to invest heavily in real localization so that people could use it and not feel like it's some Yankee imperialist thing being dumped on them. And I don't actually know who was using it but I know we sold a lot of copies of it.

Sales Video for Ross Systems Max Pro/ DEC 350

Ross: If you guys want to watch this video?

Johnson: Sure. Let's just see it.

<video playing>

Video: "There are a number of divisions which use several different micro or mini computers and software packages. That was no trouble for Fred when he worked at the divisional level. But now, he needs more flexibility, and all of those systems and programs are making his job more difficult than it should be. Fred's only option is to go to the data processing manager to see if he can get some time on the corporate main frame."

Johnson: Did you contract with a professional firm?

Ross: This was professionally done, yes.

Johnson: This guy probably graduated from Hill Drama Department, right? <laughter>

Video: "And the system will be tied up for the next 3 to 4 days."

Ross: Is that a Wang in the background?

Johnson: What year was this?

Ross: 1983.

Video: "It's out of my control. I've got to find a way to get control of my data. I can't go through this all the time."

"There must be a way."

Johnson: This shows how quickly these things age.

Video: "The system. It sounded interesting when I spoke with our salesperson the other day. Let's give them a call."

"What the Digital Equipment salesperson was introducing was the new Professional 350 business computer, and a new software package by Ross Systems called Max Pro. Fred's company was in a familiar position. They had plenty of raw computing power, a large mainframe, several minicomputers, and an assortment of micros. But they lacked a single unified system of handling financial data. Standalone micros are useful but limited in scope."

Hines: That was our real way to get financing.

Video: "And the spreadsheet software of the microcomputers doesn't offer the flexibility that Max Pro provides."

"Because of these limitations, more and more companies are looking for a way to link micros with host computers, and now there is a way. Ross Systems' Max Pro and Digital Equipment Corporations' Professional 350. The Max Pro software package was designed specifically for the state-of-the-art DEC 350 computer. Max Pro is a distributed financial planning package designed to meet the needs of individual users and company wide teams and everything in between. Max Pro combines the personal computing functions of standalone systems with the powerful computing capabilities of host systems. Even with its great power, Max Pro is simple to learn."

Johnson: Great graphics. <laughter>

Video: "English and simple question and answer commands, and online help is always available. Max Pro is equally effective for the executive or manager who has little time to learn programming and the financial analyst who maintains corporate systems. By next quarter, Fred's company had installed a complement of DEC 350s utilizing Max Pro. He and his colleagues saw how Max Pro could go far beyond conventional spreadsheeting."

"Let's get started and pull together the elements of the budget update."

"Some of the items that we're going to forecast are already loaded on the VAX."

"All my division budgets are already online with 350. Let's pull that across." <click, click>

Kapor: Across what? <laughter>

Video: "And, if the hour of a host computer is required, all the models and data can be transferred at a flick of the switch to a central fax system running the..."

Johnson: Flick of the switch of the tape, huh?

Video: "Which is fully compatible with Max Pro."

Haigh: Great cases.

Video: "Our sales managers are optimistic, but I'm not convinced. What if their forecasts are off by 10%?"

"Let's run a model on that"

"The executive staff is looking to make ten million dollars in net profit after taxes. Let's run the model backwards and see what our sales targets have to be."

Kapor: …or earnings per share? <laughter>

Video: "...user. Goal seeking and sensitivity analysis, unlimited logic, instant consolidations, integral business graphics, an extensive library of pretested financial functions, and a built-in financial database.

Man 7: What year was this, Ken?

Ross: 1983 or about.

Video: "Separate models and forecasts can be generated."

Ross: Actually sort of relevant.

Video: "Every division can have access to common information, yet every user can work independently. Models and reports generated by... "

Ross: There you go. Little plotters, you know.

Video: "Proven in use by major corporations since 1975."

"Since we started using Max Pro software, our planning and budgeting comes much faster, and a lot more productive. Max Pro gives us so much flexibility, we're able to work on everything from the biggest, most complex projects to the simplest details. And I never have to wait in line."

Keet: So where do I sign? <laughter>

Video: "...the world's most recognizable names. Our consulting experience combined with our expertise in hardware and software..."

Ross: About 20 megabytes worth on that old system.

Video: "In addition..."

Carr: Push button phone, no rotary dial.

Video: "...hotline service answers questions as they arise."

Keet: VT100.

Video: "Our comprehensive instruction materials and documentation assist users. Today, Ross Systems is one of the fastest growing software and remote computing companies providing financial management systems and computer services to clients in a variety of industries worldwide. Ross Systems and Digital Equipment Corporation, setting the standard for ..."

DP Managers' Issues

Johnson: What I want to discuss is a project that Bill Aspray actually was involved in. We could sit here and giggle because it looks outdated. But one of the most interesting research projects I've heard about is where Bill and somebody else did an analysis of issues that data processing managers were facing. What they did was to go through the ads that computer companies ran. Tell us a little bit about that. They were able to do an analysis of what it was that data processing managers were concerned about at that point by reading the ads that had been run, reviewing the ads.

Keet: Well, that's pretty much it.

Johnson: What year was it?

Keet: We looked at a period from 1950 up to about 1980, at every five year intervals, looking at ads in about 25 publications ranging from business journals to tech professional journals to very popular mass market things. And one of the problems that we weren't able to control for was the rapid improvements in the advertising industry. But you could read lots of trends that were going on over this.

Johnson: So it's interesting that artifacts that are ephemeral like this can really be useful.

Carr: Can be very useful. Obviously we were all tripping on that. And we didn't even see what the selling hot points were.

Ross: And actually some of them are still sort of relevant today.

Carr: Just change the haircut.

Kapor: It would be very interesting also to interview people who were the target audiences for those, independently of this, to see what their recollection of what they thought their problems were, to see if there were systematic ways in which the pitch is either met or missed -- the customer perception.

Ross: The other thing I remember, you saw the logo, the spinning Ross Systems logo with the dot in it. This is funny, but five years later, we hired this marketing consulting company, and paid them a lot of money to do an analysis, and they came to the conclusion that we had to remove the dot from the logo.

Phillips: That will cost you "X."

Kapor: And that's why they bring in the big guys.

Haigh: And you paid it.

Ross: We paid it. And the dot was gone. I remember some sales guy walking around with this thing glued onto his forehead with the dot.

PC Response Speed

Haigh: I saw how slowly the text was appearing on the terminal. Did it really do that or did you just do that for the video?

Ross: That was the speed in those days. We had to work on the DEC Pro 350 which was an RSX-11M based operating system, so we were working with 64k.

Kapor: Yes, we ran into this because it was...

Ross: Overlaid and...

Kapor: End users, sitting at the device, like it when the screen updates immediately and imperceptibly. But there was a period of time in which this was not known and accepted because you had contention between less powerful standalone PCs that had that behavior, and more powerful computers with timesharing terminals, but they had like 19.2KB, and they would take an appreciable period of time to paint the screen, so you couldn't do a modern spreadsheet where you change a cell, and something updates right away. I was a total PC bigot then, and probably still am one now, only slightly mellowed. It just seemed obvious that you had to have the processor, and the memory, everything you need local to you so you didn't have to paint the screen in this slow and painful way. But it was a very contentious issue.

Ross: There were a couple of VAX spreadsheet programs.

Kapor: Yes. And I said, "These are doomed."

Hoxie: One second. We were painfully aware of that, so we did a trick. I think you were the one that actually did it. We would compute the cells off screen with little timers saying "we're still computing." And then when it was on that screen, it was already moving into the screen buffer so it appeared like that, so people wouldn't see the cell going click, click, click, click. So we had masked the slowness by doing that.

Keet: Yeah, I was at the same point. We were still selling through National CSS timesharing in the early 1980s, of course, and you've got to remember that people were paying the price of a PC each month in a typical bill. A typical National CSS bill was like $3,000 a month per terminal. And the compute time, these were on the old golf ball Selectric typewriters. And an anecdote, which I think is very interesting, was that they were so slow in getting back with the results because the mainframes were slow and there were so many users. Hal Feinleib, who was the author of VP/CSS, had so many complaints from the users that they thought their line had died or the machine had gone down, that he invented this little tick so that the golf ball would go Ch-ck <sound>. And that kept the National CSS users happy because they knew that there was something at the other end of the line. That's how slow it was. So if you combine paying $3,000 a month and waiting for a Ch-ck <sound> you can see why the PC and PC applications took off.

Kapor: The interesting thing I think for the historians, and we have historians here of multiple generations it appears <laughter>, but many issues become so definitively settled that if you weren't there, or if you're not a historian studying this, you don't even understand what the contentious issues of the day were. I think it is important to recover what these issues are because they're all very instructive. And as we were talking about before, many of those issues appear today simply in another guise. And the only thing that gives us 50-somethings and on up any possibility of being competitive with today's 20-year-olds is not, "My fast ball is just a lot slower," but understanding that these issues are not new and you can make good decisions now if you know it. Well, you can beat the crap out of these 20-year-olds who don't think that history matters. <laughter> They'll start anything, you know, it's great.

Hoxie: Hey, don't give our secrets out. Come on.

Ross: I just looked at this restoration of the 1401 that they’re doing downstairs [at the Computer History Museum], which is the first computer I ever programmed on, and you know, the issues are different, but not different going back as far as 40 years.

Hoxie: You just don't have to set word marks anymore.

Ross: You don't have to set word marks anymore, right. You remember that, huh?

Hoxie: Well one of my first jobs for IBM in 1964 was upgrading 1401s down on Wall Street mainly for printer drivers. But they were selling 16K memory modules. Anyone want to hazard a guess what a 16K memory upgrade to a 1401 cost?

Ross: Huge.

Kapor: $1,000

Ross: $64,000?

Hoxie: $60,000.

Ross: And that would be in 1964.

Hoxie: Those are real dollars. At that time, as a systems engineer, I was − I'm very embarrassed to say this − I was making $5,500 a year.

Cortada: To put that in context, I became an IBM salesman in 1974, in the Data Processing Division, and I made $16,000 a year. To put that in perspective, a newly-minted PhD historian that year would have made $11,500. That's why I went to work for IBM.

Keet: When I moved out of engineering and took a sales territory with IBM, to put a capper on that one in 1966, my base salary was $425 a month.

IBM as a Source of Industry People

Hines: How many people here worked for IBM at some point?

Ross: I did, for one summer.

Cortada: So were you a systems engineer, or were you a peddler?

Keet: I was an SE.

Cortada: You were an SE first.

Keet: But I dropped. I went backwards. I became a salesman.

Ross: I was a summer intern, programmer.

Hines: I was a programmer-analyst. $8,600 a year to start.

Cortada: Plus you had all the benefits. It was a very paternalistic company.

Hines: But we couldn't wear pants.

Hoxie: Besides Robert, did anybody else work in Xerox? Because I think IBM was a great source of people going out and doing creative things. But Xerox was also. Are you the only one here who has a Xerox background?

Johnson: Interestingly, the trade association that I was involved with for years as was Lee which was the mainframe software services group which was heavily populated out of the IBM culture. And then the PC guys came along and they went off in a whole different direction and formed different connections. As recently as about 1993, 1994, Howard Anderson of Yankee Group was giving a presentation at what was then an ITAA conference, and asked how many people in the room worked for IBM at one time. I think there must have been 90% of the people who raised their hands. That company really populated the mainframe software services industry. That's where everybody came from.

Cortada: Well, that's just happened several times in business history. We had the same situation with NCR and Burroughs, in the pre-computer era. In fact, NCR provided probably more CEOs to other companies and other industries.

Kapor: There's always a largest employer or two or three employers. So in the Silicon Valley in all the software companies you see tons of Netscape people there, because at one point, they employed everybody.

Cortada: IBM had 375,000 employees in the late 1960s.

Ceruzzi: What do they have now, probably half?

Cortada: 330,000.

Ceruzzi: So it's getting back up.

Cortada: Well, even then, I mean, the number of people on the actual payroll in 1989 when we started really turning over people, who were still on the payroll in 1985. It was only 25%.

M7: That's a lot of alumni.

Cortada: But unlike Digital Equipment, the company survived, which is kind of the difference between them.

Keet: That's why we have gray hair.

Phillips: And so now, because it brought in the marketing person...

Carr: Lou Gerstner.

Phillips: Who came from a food company, right?

Kapor: RJR Nabisco.

Keet: But he was from McKinsey & Co. Consultants, that's where he was from originally.

Carr: Let's track a little bit of genealogy -- who came out of IBM and played a role in the PC software industry, either inside of IBM or externally?

Cortada: A lot of folks from Compaq came out of that.

Kapor: The founders were out of TI, but they hired a bunch of IBM people.

Cortada: Well, Digital brought in this guy from TI.

Keet: Actually, inside IBM there's a whole story that should be told. There was a big internecine war over the PC, and finally it was Frank Cary, as I recall, who -- he was ex-CEO and Chairman of the Executive Committee at the time -- put his personal stamp on Don Estridge. I don't know how many people remember Don Estridge. He was a great guy. He and his wife were killed in the Delta crash in Dallas because of wind shear. And I think it would have changed the course of IBM had he lived, because he was really in the know

Kapor: He already had been removed from running the PC division before he died in the crash. And they put in Bill Odin. They put a company man to rein it in. So Don might have bounced back, because that can happen. But his star was on the wane.

Cortada: Yeah, Bill Odin had taken over that division. He grew up in a traditional IBM DPD, large systems environment.

Ceruzzi: They had this attempt to do the same thing with a 3270/PC, where they wanted to have the best of all worlds, which was the worst of both.

Carr: So, did they provide much talent into the software companies, though? I'm coming up with a dry well. I mean, I'm thinking IBM's influence was all the research inventing the RISC processor, the Winchester disk drive, SQL, relational databases, you know. That ultimately has had a huge influence, but not PC software, not key entrepreneurs or business managers, it would seem.

Kapor: Well, a lot of CEOs, and even more VPs of sales, have IBM backgrounds, and that's the big spawning ground. But in the PC industry, you see IBM.

Ceruzzi: But not programmers.

Keet: We used to have a saying that you never hire an IBMer directly from IBM. You always hire them on the first bounce.

Cortada: Always wait 6 months.

Johnson: Let them go somewhere else. That's from the memoir, the interview that we did with Lee Keet, when he was talking about coming out of IBM.

Cortada: And everybody knew that, and probably the IBMers knew that they had to go through two jobs before they found what their calling in life would be outside of IBM. So it was folk wisdom but it was understood by both parties, and not just in the United States.

[End of Session]