An Interview with

DAN BRICKLIN AND BOB FRANKSTON

OH 402

Conducted by Martin Campbell-Kelly and Paul Ceruzzi

on

7 May 2004

Needham, Massachusetts

Charles Babbage Institute
Center for the History of Information Technology
University of Minnesota, Minneapolis

Dan Bricklin and Bob Frankston Interview

7 May 2004

Oral History 402

Abstract

ABSTRACT: Dan Bricklin and Bob Frankston discuss the creation of VisiCalc, the pioneering spreadsheet application. Bricklin and Frankston begin by discussing their educational backgrounds and experiences in computing, especially with MIT's Multics system. Bricklin then worked for DEC on typesetting and word-processing computers and, after a short time with a small start-up company, went to Harvard Business School. After MIT Frankston worked for White Weld and Interactive Data. The interview examines many of the technical, design, and programming choices in creating VisiCalc as well as interactions with Dan Fylstra and several business advisors. Bricklin comments on entries from his dated notebooks about these interactions. The interview reviews the incorporation of Software Arts in 1979, then describes early marketing of VisiCalc and the value of product evangelizing. There is discussion of rising competition from Mitch Kapor's 1-2-3 and the steps taken by Fylstra's software publishing company Personal Software (later VisiCorp). Part II of the interview begins with Bricklin and Frankston's use of a Prime minicomputer to compile VisiCalc's code for the Apple II computer. There is discussion of connections to Apple Computer and DEC, as well as publicity at the West Coast Computer Faire. The two evaluate the Fylstra essay, reviewing the naming of VisiCalc and discussing the division of labor between software developers and Fylstra as a software publisher. They describe the porting of code for versions of VisiCalc for other computers. Talks concerning a possible merger between Personal Software and Software Arts broke down, and instead there was a lawsuit between the companies. Software Arts developed a new product called TK!Solver and became a Macintosh developer. They evaluate the early spreadsheet products, including Kapor's 1-2-3, Context MBA, Multiplan, and Excel as well as VisiCalc for the IBM PC. There is discussion of the Xerox PARC Alto and Macintosh.

The interview was conducted in two parts, first with Martin Campbell-Kelly and the second with Paul Ceruzzi. There are references to an essay prepared by Dan Fylstra, circulated privately at the 2004 meeting and subsequently posted on the web (see "The Creation and Destruction of VisiCalc" in footnote below).

Preface

As part of the Software History Center's collection and preservation activities, and in conjunction with its meeting on the history of personal computer software held in Needham, MA, on May 7, 2004, the Software History Center (SHC) arranged for 14 oral histories to be conducted with computer software company founders and other key industry participants. All of these oral history interviews were conducted by historians well qualified by their knowledge and interest in computing history.

The following is a list of the people who were interviewed together with the name of their interviewer:

John Brackett and Doug Ross, interviewed by Michael Mahoney
Dan Bricklin and Bob Frankston, interviewed by Martin Campbell-Kelly and Paul Ceruzzi
Jerry Dreyer, interviewed by Thomas Haigh
Ben Dyer, interviewed by Nathan Ensmenger
Dan Fylstra, interviewed by Thomas Haigh
Gary Harpst, interviewed by Tim Bergin
John Imlay, interviewed by Bill Aspray
Luanne Johnson, interviewed by Janet Abbate
John Landry, interviewed by David Grier
Mike Maples, interviewed by Nathan Ensmenger
Seymour Rubinstein, interviewed by Jeffrey Yost
Jonathan Sachs, interviewed by Martin Campbell-Kelly
Oscar Schachter, interviewed by Thomas Haigh

Each interview was tape recorded, transcribed and edited by the SHC, the interviewer, and the interviewee to ensure clarity and readability without changing the style or flow. The original tapes along with the edited transcripts were donated by SHC to the Charles Babbage Institute (CBI), which placed the edited transcripts on the CBI website and have archived the audio tapes.

On January 1, 2005 the Software History Center merged with the Computer History Museum, and its work is continuing as the Software Business History Committee as part of the Museum's activities (see www.softwarehistory.org).

Software History Center Oral History Program

Dan Bricklin and Bob Frankston Interview

Martin Campbell-Kelly: My name's Martin Campbell-Kelly and I'm with Dan Bricklin and Bob Frankston. It's May 7, 2004 and we're at the Sheraton Needham Hotel in Needham, Massachusetts. This is part of the Software History Center's Oral History project. I'll just start with some general questions about your background.

Dan, you have an entrepreneurial family background. Can you tell me how this shaped your career?

Dan Bricklin: My grandfather founded his own business. My father worked in that business; it was a small printing business and I always wanted to start a business both as I recall and as my relatives remind me. If we found some money on the ground or something, my cousin said he would go off and want to buy ice cream with it, and I wanted to start a business with it – or something like that. So I've always wanted to do that.

I think part of that is being used to having a small business and not depending on anything other than the business so that you're not afraid to have your own business and the problems thereof. Despite problems and things like that, my folks always made me feel like it was OK – I think that helped.

Campbell-Kelly: You went to MIT, where you majored in computer science. Can you tell me something about how that education experience influenced VisiCalc and Software Arts? For example, I believe that you developed an online calculator when you were an undergraduate.

Bricklin: At MIT I started out as a math major, but wasn't doing anywhere near as well in math as I was in computers, so I switched in the middle – so my grade point average showed quite a bit that math was not my skill.

The first semester I tried to stay away from computers for a while, except taking one class: 6-251 which was a class teaching you about compilers and computers, etc. But I remember going into the final I was #1 in the class. So at that point I decided to get a job in computers and I went around to Project Mac and at Project Mac I stopped in a lot of offices and when I stopped at Professor Corbató's office, he said, "Oh, we can give you a job maybe working on Multics." And others were working to get credit, but I asked for a part time job and ended up working under him.

When I started working on the Multics project, I guess the thing I figured was – I looked at it and there wasn't a calculator available in Multics and I had just learned a bit about this type of stuff in the course. I think it was the night before the final; I pulled an all-nighter and wrote a program called Calc that let you say "CALC 2 + 2", hit return, and it would say 4. It was not done very well as I understand.

But that wasn't the background that did it. One of the things was – that's where I met Bob Frankston – at Project Mac working on Multics. My first project was – I was told to finish up something he had done as his Bachelor's thesis. I had to actually do a little bit of cleaning up to make it fit with the newest version of the system – with providing computers to people, which is what his thing had to do with.

I ended up working on the human interface side of Multics. I was on the command system – that was the group I was in, not in the base operating system part. And I ended up also implementing interpretive languages. I ended up on the APL project, implementing APL actually. We did that twice – once working for somebody else and then the second time to make it faster, learning how to make it more efficient. I did it with the group that I led. So those are two interpreters which is real important.

I also did a memo program where you type a memo and a certain number of dates; tomorrow an appointment with such and such. I wrote a program called Convert Date to Binary that would let you type the dates in almost any format and it would convert it to binary – the date format that Multics had. It kept the calendar. So for Multics, I was into this user interface. This was all the software side. But my background there was working with really great people – being exposed to all these people I ended up with: Bob, David Reed, and people like that.

The internet was just starting in those days. A friend of ours was building the interface message processor [IMP] for Multics to connect to the Internet.

Campbell-Kelly: Just remind me of the dates, then.

Bricklin: I came in 1969. It was my freshman year in the fall of 1969; I graduated in June 1973. I finished the Multics APL, the second one, that summer. So it was interpreters, which is real important; Multics APL was a numeric language, but I was also doing a lot of human interface stuff.

And then there were courses that just taught you general things like the Systems course 6-233 it was called in those days. So that was real important – to learn that type of stuff.

Campbell-Kelly: Bob, tell me about how you got into MIT. I think you did graduate work with Fernando Corbató.

Bob Frankston: Actually the entrepreneurial background is similar, a little different than Dan's. As a kid, I would be helping my father by packing TV parts in boxes and we'd sell them at Radio Shack and companies like that. The idea of working for somebody . . . it wasn't just that it was OK to be an entrepreneur; it was why work for somebody? It seemed to be a strange concept. I would do it in some way sometimes, but I was never really an employee in my mind. So that was the background, that you'd always just do things on your own. If you worked for a company, you were rarely on your own. Maybe it's an ADHD thing, but I think the programming sort of aided and abetted that. You could do things on your own that were significant.

One of the stage setting things, even before MIT, was that it was a new field. You can use terminology "no barriers to entry." So even in high school, one of my jobs was one job converting an IBM software package to the 1130 one summer. The next one I started working on was an online information service with White Weld in 1966 (since then I realized the work we did played a role in creating today's "quants"). I'm working with people like Butler Lampson, who later was one of the founders of Xerox PARC. So that was just the things a kid would do casually. I learned to write compilers just for practice.

That job actually continued while I was at MIT, so I had one foot in the business, the financial information world, programming online services – as well as not so much the mainframes as much as large interactive systems external to MIT. Every other time I applied at MIT I'd get in. So the first time I got on the waiting list, I went to Stony Brook for a year and wound up starting out with a graduate class in computer languages.

Campbell-Kelly: Tell me the dates you were at MIT.

Frankston: Okay, first year I was at Stony Brook from 1966 to 1967 in New York. Then I transferred to MIT in 1967. I was going for two degrees, but then once I got a high draft number, I just graduated in math and what they called a computer science degree, but it was an experimental program then, otherwise known as 6-3.

In 1970 I applied to grad school; I didn't get in. I reapplied with Corbató's recommendation and I got in the semester after. I got my Master's Degree (and an Engineer's degree which the Europeans would know about) in early 1974, and then it was suggested that maybe graduate school was not my future in 1976, but I still stayed friendly with MIT.

Campbell-Kelly: What kind of work did you do with Multics?

Frankston: During the Multics project, probably 1968, it was basically the early stage of the machine – it was just becoming available. I did a Bachelor's thesis on the Limited Service Subsystem which Dan then implemented.

Bricklin: I finished it.

Frankston: I remember writing about it; I don't remember whether I coded it, but I know Dan actually had to finish it up and make it work. I later used it to create an online service for students.

Then in graduate school I did a number of things. My Master's thesis was supposed to be on measuring resources, but instead I said – who cares about CPU cycles, you care what people understand about the computer.

When I was an undergraduate I helped co-found the [MIT] Student Information Processing Board which made computers accessible to students, because my experience in Online Services – I'd always been concerned with how people actually use the computers, as well as the technology underneath. So there's a whole sort of systems – I can get out a soldering iron if I have to. I can build a time-sharing system. But what I never lost was that the ultimate goal was usability.

So therefore my Master's thesis basically became a thesis on micro-transactions. How do you present this in ways that make sense to users and also I assumed by then a worldwide network where computers reconciled transactions. We used the term "Information Utility." Today it would be called Cloud Computing.

As a matter of fact, my doctoral thesis was going to be about a distributed database – everybody would have their own personal computers connected up with networks around the world. But it was back in 1974.

But I only wanted to implement it; they wanted me to write about it, so I spent time instead in the clinical decision-making group, but I was always interested in AI, linguistics, accounting, ambiguity – all those kinds of issues. And computers basically were an applied philosophical medium. But at the same time, the trick was to always be implementing systems for people to use and I would tend to do things for me to use, but in a way it was useful for other people. Like when I was at Interactive Data, first I was working on this First Financial Language. Later I needed a de-bugger for myself so I wrote it, which became the IDC de-bugger.

Campbell-Kelly: Dan, tell me what you did between graduating and starting on the MBA at Harvard. According to the materials I've seen, you worked on Digital Equipment's WPS-8 word processing system. Did that influence your view on personal computer applications? What other experiences now strike you as having been formative?

Bricklin: When I graduated, the first thing I did was finish doing the APL project. That was one of the things I did that summer of 1973. There was a fortuitous event. What happened was: I didn't get into grad schools – the ones I wanted to. Thanks to my C's and D's in math; it

didn't balance out my A's well enough. Getting into Stanford, Carnegie or MIT for graduate school in computer science was not in the cards.

So I ended up looking for a job. One of my co-workers, Paul Green, had interviewed at Digital Equipment Corp. and he turned down a job with them and said – I'm not the one, but you should ask Dan who I work with – interview him. This was for the group that was doing languages. So I went out there and interviewed with them. During lunch, I ran into my office mate from working at Multics, Michael Spier. Mike came up and said – Hi, Dan, how're you doing!? He used more colorful language.

He said, "You've got to meet Valdeane [Alusic], you've got to see my group." He was working in the computerized typesetter group. So I went over and interviewed there at the typesetting group, and they wanted me too, apparently.

Now, my father was a printer and my grandfather was a printer, and this is human interface again. It was much more interesting than doing FORTRAN compilers. So I got my offer and I said – That's great. The letter comes from DEC; it was for the FORTRAN thing. I said – No, no, no. So I got the identical letter back saying congratulations and all that. I got a job working for Valdeane Alusic who headed the group that was doing, Typeset-10. So I started work with that in the fall of 1973. I was working on wire service processing. In other words, the data is coming in off the wire services and we had a typesetting system based on PDP-10. Our stuff would take it and convert it into computer files, break it up into stories, and send it off to the typesetting system that had multiple feeds. So I worked with that.

I was traveling to see real customers. We had the Kansas City Star, the West Palm Beach Post Times and the London, Ontario Free Press. I got to actually get out and see customers in the real world where my users drank a lot, had deadlines that were real – the newspaper had never missed an edition in 100 years and you don't want to be the one to cause it! If it was 15 minutes late it costs them tens of thousands of dollars.

So I learned a lot about the real world through that. Then the second year at DEC I started working with . . . was it the second year? Well, part of what I was doing at DEC with the wire services – we also had to get our editing terminals going. So I worked on a project to get our editing terminals for Typeset-10 working. This was important because it was my first word processing-like system. This was a system that was distributed; it was a client/server system. We had this PDP-11 that had two screens on it, memory-mapped screens. And it was connected by a low bandwidth line to a PDP-10 which could not take input very quickly because if you typed any faster it would overrun the input buffers, so they could output at relatively high speed, but input at slow speed; we had to tune for that.

It worked out that we mirrored what was on the PDP-10 with what was on the screen. I brought in one of the students who had worked with me at the APL Group, David Moon, who later went to Symbolics. I hired him for the summer. He helped work with me and he helped implement one piece of that and I implemented one of the other pieces. So that got me into computerized editing.

I also learned about deadlines – that's when the Watergate tapes, the Nixon tapes were coming through and I had to quick drop out, flying from one place to another to help fix the computer to be able to read the tapes to save time and all. I learned a lot about the real world from that and about doing video editing terminals.

Out of that I ended up in the word processing group at DEC because DEC was starting to do word processing. In the meantime, waiting for that group to start, I worked in a group that developed a new terminal for the typesetting group, both for the 11 and for the 10, based on the PDP-11 (the LSI-11). That was a memory-mapped screen connected to an LSI-11 computer on a single machine with a keyboard. I helped specify it because of what I needed out of the hardware. I actually helped code the microcode for that – I wrote a compiler for the microcode. I worked closely with the guys building the thing, literally the integrated circuits and the whole stuff.

So I got to help design that. That was on par with a PC – what would have been an upper end PC of a few years later than that machine. Then I went on to be project leader of the word processing system. I was project leader because the senior programmer on the group was a contract worker and he wasn't allowed to be the project leader. I led and worked on the printing system, the file system, the boot loader, the main user interface loop – all that stuff.

So that got me into types in computerized word processing, and I learned the whole history of the word processing industry – how word processing came about which I think is important to understand. This is all well before WordPro and stuff. This is before Wang had video editing terminals. They were just starting with it. The history of word processing is fascinating. We were investigating it because we were using it as part of our marketing to show the whole history of word processing up to that point. We had to invent things like the embedded ruler, which was not common in those days. Usually what they did was simulate a typewriter and the ruler was active at the time you typed and then you would change the ruler so that the idea of page oriented versus document oriented was a major discussion and I pushed hard for document oriented, which now we all take for granted. Page oriented was more like Page Maker. Document oriented is like [Microsoft] Word.

But people thought that they had to do page oriented – we had all sorts of arguments about that. And I wrote the actual spec for the DEC word processing system which was sold for, I think about 10 years. That was built on a PDP-8 system with 16K 12-bit words. There I got a lot of experience developing user interfaces. If you take together the typesetting background, which was keystroke minimization – because people were paid by the keystroke – and put it together with word

processing, where also people were paid by the hour or by the keystroke. We realized that keystroke minimization was important. Efficiency of using it and stuff like that was very important. Learning it, making things understandable – you're working with unskilled people or people who are under pressure, so we were aiming for that and worked hard with people who were very experienced.

The head of the entire DEC word processing project product line was Jack Gilmore. Jack Gilmore has been seen by many people in the Computer History Museum because he's in a videotape of Jay Forrester being interviewed by Edward R. Morrow and he's in the background of that. He had pulled an all-nighter to do the demo that was being used by Jay Forrester to show how the Whirlwind computer was being used.

You could see Jack. And Jack, in 1951, in that videotape, was the same age, I believe, that I was when I worked for Jack. That videotape was done the year I was born. So I felt connected to the past and that's why I always felt this and that's why I was always looking at history and stuff.

The keystroke minimization – also I was struck with that when I was in the typesetting group. I remember the head of one of the newspapers came to visit us and said – We're just a poor newspaper; you're the big Digital – but listen: your product's a little late – we've got to get it soon because I have a deal with my union that I can lay off as many people as I want until next September. After that, everybody is hired for life so your products better work before that because I don't want to miss any editions!

And I learned all these things about the real world and about how – well, if my thing is more efficient, then I can do a better job. There were a whole lot of other things I learned – I learned so much from that whole process. Finally, some of the ideas of VisiCalc came out of that. One is the idea of the scrolling with locked titles was one thing we had talked about in our group and we were talking about working with tabular material. We never implemented that.

The other was – I was exposed to the Harris 2200, which most people don't seem to remember – I have a write-up from Seybold of that Harris 2200. The machine was very interesting in that it was for creating display ads for typesetting. And what it had is – when it comes to the interface I think ... you had mentioned, Martin, that the ease of use and the understandability of VisiCalc was extremely important about the spreadsheet. The design of the user interface had certain things that affected it.

The design of the word processing system at DEC was the model that I used in doing it, and the specific ideas that ended up making it possible to simplify it came out of the things that I'm talking about. So if, for posterity, we care about the specific influences – because I like to credit people where credit is due – I think that the things that came into it, people should understand this building on others and why it's important to be exposed to ideas – going to things like museums and stuff

like that, and seeing what people tried to get inspiration for ideas that maybe are more appropriately used in a new product.

So the thing about the Harris 2200 is that it was designed for display ads – so you had pieces of type at different sizes and all, and you could use a track ball instead of a mouse – you could move around to select a particular item and then you could use the track ball to do it. It had a list on the side of the information about each of these pieces of type. So basically it had the idea of . . . there was this status line . . . actually a lot of lists of status lines, that when you selected something the status line highlighted and the status line would say: this is the font; this is the point size and this is the location. So there was this concept of this little thing and that inspired me when it came to the designing of the status line and solving the problems I had to solve when I came up with the VisiCalc interface.

But that was at the end of working for DEC. I left DEC. So the machine was basically on a par with the early PC machines that we used – the PDP-8, which was a very slow, simple machine, 16K 12-bit words, 100K floppies. Operating system – everything fit on it, but it was a screen that was not a memory mapped screen. It could only scroll up and down.

We also had to teach Diablo how to make their printer work, because their printer wasn't very reliable . . . it was a daisywheel printer. The DEC manufacturers taught them that and I learned about a lot of that stuff.

DEC moved the office up to New Hampshire and so we all had to move – either you were with them or not. Being single, I didn't want to move up to New Hampshire where there wasn't much of a social life. So I decided I was going to stay down in the Boston area so I started looking for another job. I ended up getting another job – it turned out to be a mile further north into New Hampshire, but I didn't have to move.

I worked for a company called Fasfax that was producing electronic cash registers for the fast food industry. They used a local area network; they called it the Blue Wire [the cable was blue coax], because this was for regular users. It was based on the Motorola 6800. It had the latest memory chips. They had used outside contractors to help develop it; it was written in Forth. I was hired to be the head of the programming department at the time. I basically took over a program that existed and it enhanced it.

We were working with users that were really low level – I mean these were kids stoned out of their minds who were flipping burgers type of thing! Our system would call each of the stores and download the latest sales information and stuff and new recipes of everything, every pickle! And sometimes it was the single store pay-phone that was used. I learned the realities of the real world which is part of my business education which was different than had I gone into compilers.

They would call the store and the store would then flip a switch on the pay phone so that we could connect the modem to it. Sometimes you'd call the store and they'd say – Sorry, we're busy; we're being robbed! I mean, it was very real world-y. But there I learned – it was a very small business, and I was amazed that here was Digital, a billion dollar company.

Campbell-Kelly: Dan, can I stop you a second? At the rate we're moving, it's going to take 8 hours to do this interview. If I do this signal – could you come out with the salient points and then we'll pick that up later. OK?

Bricklin: OK – the thing that I got from Fasfax was – it was a small company and yet it had all the things Digital had. It was doing the same hardware at the same level. It was doing micro-processing; I was using the latest microprocessors in assembly language as well as in higher level, in Forth. I was doing assembly language code on a microcomputer. I was amazed how the small company could do all the things – sales and I saw all the parts of it run by some brash guy who was running the company. And that impressed upon me about a small business and about working with the microcomputers, and that was part of the experience I had with that. In the middle of that, I applied to business school and got into Harvard Business School.

Campbell-Kelly: Bob, this question is for you. After leaving MIT, explain what you did.

Frankston: First, in listening to Dan, I'm not going through all the influences, but I'm realizing that there were a lot of influences throughout. Even before MIT on the SDS 940 there was QED, which is an editor, but the version we used – unlike the versions later – this is the era of full duplex teletypes where the character typed on the keyboard and the character printed had no relationship. As a matter of fact, when the New York phone lines were busy, I'd call the Los Angeles phone number and the characters would echo a couple times across the country, so I could completely disassociate. So you learned really about the separation between the presentation and how the keyboard worked, especially in the teletype days.

Harris *[corrected below: Hendrix]* was a company in New Hampshire doing things and they were using some of the same things and I actually spec-ed the keycaps. Why? Because we were looking at basically the screen version of the editor, and those kind of capabilities.

There were various typesetters, but one of the influences – the key one actually – was Jerry Saltzer's Runoff, which eventually led to this thing called HTML. This was the whole issue of separating the presentation from the data. One direction – some of the QED influence – was with TECO. There was MIT TECO and later there was a DEC version that had another interactively type control keys – the same as QED. And it would be influencing sort of what appeared on the screen – originally the teletype, later the screen, not directly.

That led to EMacs – one branch. We really disassociated presentation. There were languages like LISP, Macsyma, and various representations. There are a lot of different influences throughout. As I mentioned, one thing I did was create an interactive system for students so you could just walk up to a computer and just write programs without having to pre-register. I wrote it originally because I wanted to practice writing a compiler and that took a night. Another night I put a front end on it. And the third day I made it a service; the fourth day it was censored, but it became a good service – very popular.

Campbell-Kelly: What jobs did you have?

Frankston: So the work at White Weld – that group formed the basis for Interactive Data. I stayed officially till 1979. But it was much of my own schedule. At the same time I was in grad school doing a research assistantship, helped David Reed as a matter of fact on his Master's thesis on improving operating systems, which to this day people still don't get.

I won't go into how bad today's operating systems are. And also in class was this guy Bob Metcalfe sitting next to me talking about putting a lineout on coax [type of cable later used by his Ethernet]. David Reed later did more networking. So there were all these different influences.

One job I did was a consulting thing but pretty much full time at a company called ECD. I was doing a 6502 based microcomputer system, multiple processors, and large memory. So I wrote a Basic for them, since I had practice writing such things. The company itself made a little mistake – manufacturing for more than it cost to sell the product. But I learned a lot about the 6502 and tools, so there was that influence.

I took some time off to go to Nigeria with my father and see if he could sell anything there. So there was this mixture of entrepreneurial and experimenting as well as Interactive Data, where I was creating real services for real users, but had a lot of flexibility.

So putting all that together – when Dan said he had this cute project – it was very interesting.

Campbell-Kelly: Dan, tell me the reasons for starting at Harvard to do the MBA?

Bricklin: When I was in college, I remember thinking that I was going to treat myself after a couple of years and go to business school. I remember I thought of that. But later on, a friend of mine's father was a laid off as a programmer in his 50s and I was sort of struck by how people like me – the young, just out of school, people – who would work for less money and would work for a million hours helped kick him out.

So I figured I would go to MBA school eventually, but in the middle of interviewing about jobs, I think one of the headhunters said that having an MBA would be helpful and I'd been planning to do

that, so I figured maybe I should do that – I'll apply. So that sort of got me into it. I chose Harvard because I visited Harvard and MIT and I liked watching Harvard's case method. A friend of mine from MIT who went to Sloan School said his friend who also went to Harvard said that he got a different set of contacts out of Harvard. And I had been so helped by my contacts that I had gotten – the network I got out of MIT – so to have an additional set from Harvard sounded like a wonderful idea. So I decided to go to Harvard.

Campbell-Kelly: We know the story of how you were exposed to the idea of spreadsheets and VisiCalc – Bob, can you remember the moment when Dan first explained this idea of a spreadsheet?

Frankston: I won't say the moment, because I wasn't thinking of a spreadsheet. One of the other influences I've had was with interactive displays was the Spatial Data Management System at the Architecture Machine Group which later became the Media Lab at MIT, so I was used to people showing screens, pointing things out – all those scenarios.

So my image, as Dan explained it to me, was you hold up this calculator – he used a TI, except as I explained yesterday, he was matching a trackball, I was just assuming IR and you point and found the direction and keyed in what you wanted. It was a much more flexible general model. The key thing that helped me see it was the way he actually reduced it to the simplicity of the grid in his Basic demo. That really made it so it much more reduced to practice. But more important, it wasn't just a simplification; it really gave the user the necessary structure.

Campbell-Kelly: You mentioned a Basic demo. What was that?

Frankston: Dan wrote a version in Basic on the Apple II to test out the ideas. That really gave the insights for what it was. One general problem you have in describing an idea is: sometimes you have a problem getting people to see the full vision and sometimes you have a problem with the opposite: the vision is there except how to reduce it to practice is hard to explain.

Campbell-Kelly: Dan, tell me about the about the invention of VisiCalc and the early prototype.

Bricklin: I had this general idea. As I started out I was always driven to try something and to actually implement it. And I think through coding it, because I work out my problems. As Bob said, you get your vision, the hard thing is reducing it to practice.

So what I did was that I scrounged computer time – people have to understand that in those days to get computer time, you had to find it wherever you could. Usually you couldn't get it unless it was a big institution and you steal or borrow it from. Being at Harvard I had access to their time-sharing system, a PDP-10. There's even a picture in our yearbook of me typing there on one of those, you can see my pony tail way down to the middle of my back.

They had Basic running on that machine and they had the VT05 character based terminals with it, where you could do positioning or whatever and re-write the screen. So I was trying to figure out... here's the idea I had for the spreadsheet: I understood that there were, in my mind, 3 planes. What the spreadsheet does is: it takes the input, the calculations and the output and puts them all together. In traditional programming at the time – FORTRAN, Basic, and stuff like that – you had input statements, you did processing statements, and then finally at the end you did formatting.

It was always the formatting stage that was the hardest thing in FORTRAN – figuring out the FORMAT statements and getting them right and getting your input and all that. What I did with VisiCalc is – I sort of put it all together where I had the input, the calculations and the output all happening at the same time. And my vision was of that. It's like word processing where basically you're working on the actual output, so you're expressing yourself into the output.

I've learned since that people think in terms of the results and so you can do the input in terms of the results and that works. So I had this idea of the 3 planes, so I started implementing something where I had it switch between. You could have one plane where you'd see . . . well, first of all you had to have numbered stuff up. So I was starting to put them around places. And just like the Harris or things like that, the typesetting machines, general positioning I was thinking of.

But then I realized that you had to be able to refer to other things, so I'm going to say: this, plus this, plus this goes here – I had that image because what I was thinking most about was sum these things – take these, add them up, put the sum here and multiply it by this and put the results there.

I had to name them, so I was trying to figure out – how do you name cell values [for use in the calculation definition] in a way that's going work. Well, I could do it in the order in which you create it – maybe that was one that people could understand. I wanted one the computer could come up with. I could have you name it, but I knew that that was too much trouble. People wouldn't go to the trouble. I could allow them to name it, but I've got to give them a default where we make it up.

So then finally I came up with: why not use a grid? Do it in a grid, and I actually implemented that grid, written in Basic on the PDP-10 at Harvard in the spring of 1978. So I thought of this idea of the 3 planes. One plane is where you type your input or you see the output because I had the idea of a cell where you'd have a value or have a formula. So 21 was 21. If it's 21 plus 22, that's a formula and it gave a result, whereas 21 plus the one that's over here gave a result.

So then I had the formulas, so there was the view where every cell had its value – every cell – either its calculated value or its actual value – or the formula, and another one had the formatting information – right justified, left justified. And the idea of text was one of the values of the text. You could switch between the two.

Well, it turned out that that was kind of inefficient and it didn't work, knowing [Edward] Tufte now, the idea of switching from seeing all numbers to all formulas to all formatting just didn't work for people. So I came up with the concept of, from the Harris 2200, where basically you're only seeing the input/output and this was out of playing with this thing. You see the number, the value, but there was a status line (I don't remember where it was at the time) and the status line showed the formatting that you had chosen using the command line. The command style is very much like typesetting.

Frankston: Actually I want to make one correction, by the way. I said Harris [above]; I meant Hendrix was the company in New Hampshire.

Bricklin: Yes. So I had the status line that had the formatting, much like in typesetting, and it had the location and all the information. The formula was there. That was the concept. And as you moved the cursor around, you would see that. So that was the basic idea of having the grid so as to be able to name it, much like a map. Of course I was used to using grids because that's what we did in the numeric stuff we were doing, but I was always assuming the free form view of it and I think that's one of the things that future people have done in attempting to improve upon spreadsheet . . . all of us have tried that.

It turned out that the idea was: general placement, where things were placed wherever you wanted by how you thought, by how it worked for you, not that this was a column and a row. The columns and rows were just to make it easier to name them. A position was whatever you thought it was, so you could have little groups and whatever. So that's as far as I went at that point. That was the spring of 1978.

Then there's the story about talking to the professors and one of them saying you should talk to [Dan] Fylstra. So I called up Fylstra. I did not tell Fylstra about the spreadsheet at that point. He told me about Microchess and how you should write it for the PC – he was a real evangelist for the PC.

The PCs of that day, they didn't have disk drives; they were just starting to get them and I realized I needed something a little more than that, but by the time the Apple had a disk drive and by the time I saw that one, it was up in snuff.

I was looking for a machine to build this on, and DEC had a machine coming out which was sort of a derivative of the machine I had worked on, that I had helped build in 1971 or 1972. This was the PDT series which I had seen as a DEC shareholder. So I tried to buy that thing, but the DEC salesman said it was impossible to get information about that.

So in the summer of 1978 I decided I was going to do this as my project. I wanted to build this program.

Campbell-Kelly: As your MBA project?

Bricklin: No. When I graduated I wanted to come up with this program even if I had to sell the machines going door to door on Route 128, to companies – because this is a good way of doing planning and financial and stuff like that. And cold sales call would be "good for the soul." So I was looking for how to be able to make this thing come to fruition. So this now gets into the entrepreneurial thing.

I'd been talking to Bob about my idea. Sometime after . . . it must have been early September or something . . . I must have heard from Fylstra or I called him or Bob got together with Fylstra and he was talking about something.

Frankston: Bridge.

Bricklin: So Bob called back and said – I talked to Fylstra about you. Fylstra was always asking for new things – whatever you had, he wanted. He wanted to publish it; he was a publisher. Bob had never seen these demos or any of the stuff, but he told him I had something that had to do with finance or whatever.

As I recall, there was a need for home finance and things like that, but whatever it was, he said – I want to see it. So according to my notes it says: "Called Dan Fylstra on Sept. 22; set to see him Monday the 25th at his house at 22 White Street."

And Sept. 25th I had just gotten OMSI stuff. I'd just gotten stuff in the mail about OMSI Pascal, which would run on the DEC machine. All my notes here are about DEC Waltham and trying to get a PDT series and using my friends at DEC to be able to find out what the specs were.

"Saw Fylstra and got Apple manuals." On the 26th I was still talking about DEC. Then I saw my professor, Barbara Jackson, at Harvard. I constantly was going to my professors. She was teaching a course on financial forecasting tools and she said – Better simple stuff. She was always pushing for simpler. She was telling me, "Whatever you do, your competition is the back of the envelope. Senior executives don't do this stuff. You've got to make it simpler, simpler." She was pushing me on that. She said, "Don't do front ends; do stand alone. Don't know price at which there would be much demand, but sure there would be some at some point." If I need consulting jobs, see her next year. Make sure there's Delta percent function. Show her spec when done. That was the 27th.

Oct. 1, 1978 – "called Dan Fylstra; told him about Professor Jackson; will see him later this week." I was still looking at different types of computers. On Oct. 4th I got the Apple II reference manual for $10.50. I was trying to find places to buy a machine to do this thing. I was desperate to get a computer on which to build . . . to try and prototype this thing.

8 October, 1978 – Notebook entry, in all caps, as Apple II only did caps: DID FINANCE TEST PROGRAM in Basic at Personal Software. Dan F. liked it.

So I basically spent the weekend working on this with an Apple II that I had gotten from him. I guess maybe I sat in his office over at his place. I don't remember if that was a special weekend, you'd have to look up to see if it was a long or holiday weekend.

Bricklin: . . . on the 8th of October. This was in Basic; it was a program that had. . . I don't remember – it was probably ABC123 [grid naming] in those days? I don't remember. It was probably ABC because I was worried about parsing it to figure out how you'd get a NO [?] when you typed the thing in. I wasn't good at compiler stuff like Bob was, and I had to decide. In those days it was – are you algebraic calculator or are you postfix, the difference between HP and TI and all that stuff.

So I said it was all too confusing. I'm just going to go with . . . like a normal calculator. As you type it in, it would calculate, that's what I pushed for, plus it was easy to program. I wrote a program that had only one screen and it had all the cells and it let you actually be able to type stuff in. I wanted to do a mouse; the Apple II had the paddles and there's a story about that, but the paddles were a dial you turned, and I sort of used it to move the cursor up and left and right and pushed the fire button and it would go up and down. It had an exclamation mark and a minus sign as indicator characters on the screen to let you know which direction it would go.

But that turned out to overshoot all the time, because it turned out it isn't a cursor, you didn't know where in the range you were because you didn't know you were half-way to the end and the thing would suddenly slide over.

So I had to figure that out and I decided instead of doing it with that, to use the arrow keys. So I had an actual prototype going. The prototype would recalculate and it was basically the idea of the spreadsheet as we see it today, except for some minor changes in the UI – you had to say U to use something before you could point. You'd say 1 + U and then use an arrow.

I can go into what happened after I built that – after that day.

Campbell-Kelly: I think we have to focus on the creation of VisiCalc in this interview. It's very interesting.

Bricklin: So I had this first demo, whatever it was – I remember one of the problems with it was that it made a noise every time it recalculated. It would highlight; you could watch the cursor. When you said calculate, it would highlight, reverse and re-display and then show the updated value, the updated value, the updated value. And it would beep, beep, beep – of course, I

didn't realize that the Apple II used timing cycles to do that, so therefore when I removed the beep, beep, beep sound it calculated much faster – instead of 20 seconds to calculate, it took 5 seconds or something.

So I had this first version. It says "Dan F. liked it," so Dan must have been there and he liked it. On 12 Oct. 1978, I showed it to John Reese. John was a classmate of mine who frequently sat next to me at Harvard Business School who also was an MIT graduate. He was the one that I would ask a lot of questions about it. He liked it he would comment later. He was the one who said – "Dan, you don't need that U; you can just say 1 + and an arrow" – and I said, "yes, you're right, because it's unambiguous. 1 + an arrow must be pointing to something because 1 + 1 arrow means you're done with it; go to the next step." So that simplification makes it easy to type something in.

It says on 13 Oct. 1978: "modified program, better human interface. Showing to Professor Jackson on Wednesday and got Bob Frankston together with Personal Software on the Bridge program." Then finally on the 14th of Oct. "did a bit more work at Personal Software." So clearly, the Apple II must have still been at Personal Software and I was over there doing it. That's the end of these notes.

So this was the Basic version. What happened was: this Basic prototype was then the thing around which – if you listen to Michael Schrage, what he writes – the idea of having a prototype is what you then can talk around; that having the prototype is what people can then react to and that's what makes good development. You notice I'm constantly getting feedback from people. I'm using Harvard; I'm using friends of all sorts to be able to figure out how to improve this. Working on a command tree and there's in one of the "Programmers at Work" book [also "Bricklin on Technology" book] it shows the command tree on figuring out what to do. And this Basic program ended up being the test bed for ideas and was like a spec – completely independent; it was written in Integer Basic, I believe, that Woz had written.

But we kept it separate and thus I would have a prototype – and of course prototyping became very important to me – for the Dan Bricklin's Demo Program [for MSDOS] and stuff. But I had a prototype product, and then we'd modify that. And then Bob implemented the other one completely independently from that. In other words, the code base was completely separate so that we could work with different ideas.

At the same time I was thinking about how to implement it technically in another way. What would be the data structures and stuff like that, which was totally different than the Basic version. At the time I assumed I was going to program this thing. Bob was busy doing the Bridge program – they didn't have printers; he got a listing by taking his SX70 camera and putting these things on the screen, taking the picture, page down, take a picture, etc. It was a very expensive way to print.

So I started experimenting with reverse video for what the bar looked like and that inverted L looked like – whether it reversed or not – playing with the look of the screen. A lot of that stuff was being played with in this Basic version. At that point we could start negotiating – like now it's time to figure out what we're going to do with this thing and establish a relationship between Personal Software and whoever's going to write it, and Bob and I decided he had the time. As Bob said – he was a freelancer at that point and I was busy in school, and this was going to take a while to write.

So that's when we started getting into the business aspects. But the basic idea of the spreadsheet [was settled]. The idea of replicate, of copying it, that probably came in later that fall in discussions with John Reese – coming up with the idea of doing replicate. Because that was obviously a very important idea. There were all these decisions to be made to fill out – just "1 + this one" was the idea at that point, but the idea of how to do copying, how to do ranges – all that type of stuff – the whole actual details of user interface followed the same way I did it at Digital. I would build state machines, write state machines on sheets of paper to try to figure out how to do this and then implement it.

Campbell-Kelly: Let me switch over to Bob. Tell me what your recollection is now of the division of labor between you and Dan.

Frankston: Okay. Dan had done the prototype as he said. I did the Bridge conversion from Integer Basic because it was just a way of getting my hands wet in dealing with the Apple and with Personal Software.

This [is] something you often do – you go for the long term. If I learned something doing a project, that has value in itself. So I started working in late November of 1978 on what was the real VisiCalc. Basically we were sort of working it out as we were doing it. Of course I worked at night, which was cheaper. One nice thing about my apartment in Arlington, Mass, was I had this huge attic. So that became a nice work area.

The division of labor was crucial because one thing I realized is – having two people whose skills overlap a lot is important. So Dan could experiment with the user interface because he had no stake in the actual code. He could tell me to fix it; that's easy. It wasn't like there was this clueless MBA hiring a programmer. He actually could write a lot of the code. So he was appreciating that I could push back, but he also could assist on the user interface change and the reverse – I was very concerned with usability.

I was always concerned – what's the field. As I've written on the VisiCalc web site, every little decision like – when you pressed the auto repeat and you lifted a finger, it had to stop. I don't care what the computer tried to do; everything had to do with the user perception.

I had an idea – we always called it re-display. We never called it display; it was re-display because we were assuming it was a continuous process. I was still doing some things with Interactive Data at the time and I remember sitting in this class – they were trying to teach us something – I have no idea what it was, but I would sit there and work out the data structures because how do you fit it in this tiny machine?

So there's a lot of thinking, over-engineering. You had no idea how fast the display was, so there was a lot of concern about that. At the same time, there was this rule: if you couldn't explain it to users, then you fix the program. It was sort of balancing all these things and at the same time, the only way to work out what the program should do was to write it and hope that you didn't have to throw away too much code because the more you wrote, the harder it was to make a fundamental change.

Here are a couple of examples. First, do you label the rows with the letters or do the columns [get the letters]. Typically, you would put the letters down and the numbers across. We reversed that because we figured in order to do time series you had to have at least a year, which is a little over 52 weeks, but you have a lot more rows. So you had fewer letter to use. It was all these . . . I'm not going to repeat all the things, but every little decision we had to make – how do you reverse things, etc., at the same time you're doing the code.

So in the original code, for example, you were able to re-label the rows and columns. Later we figured out we needed a stable reference. We couldn't support a product where we couldn't name the cells, if we're supporting on the telephone! So it forced the cell, row, column label to stay there; you could no longer go into them, but you could put labels below. Again, each decision we had to make at the same time we were was coding.

At the beginning – later the program got ahead of the prototype – the other important thing in order for that to work was that the program had to always be useable because how could you learn what the program should do if it's not doing it? So I rapidly implemented enough to do – now division is hard but at least I could do integer multiplication and division. So I stubbed out the math by putting in enough so Dan could start using it in class.

But the programming – the way you do good programming is you keep it well structured and flexible so I can understand it. Because basically it's hard to figure out what a program does after you write it. One of the lessons I've learned is – I'm not going to remember why I did this, so I'm going to be just like any other clueless person reading the code!

Bricklin: Yes, I want to comment on this. Bob pointed out – a key thing was to make the thing describable. He jumped from when we had the prototype in October to when we had made the decision to do this as a product. I made the agreement with Fylstra later in the fall of 1978, and

then us deciding to do the implementation. The implementation started around the end of December.

Frankston: End of November, actually.

Bricklin: The thing is – we had to go rent a typewriter for me to write the documentation with. What I would do is: I would write the documentation; I wrote a reference card that I kept up to date constantly. I have copies of it here but each one had a date and it was basically what became the reference card. It said: this is what each keystroke does; this is how it works. And I started typing a manual, teaching it, which basically was a description of how to use it, but it also was a spec for Bob to work from.

If it wasn't easy to describe, then we would change how it was implemented. These evolved over time, constantly doing that. So there was this back and forth. In addition, I helped Bob with the design of some of the data structures for storing the data. But as Bob mentioned before, his background was in this re-display, this interaction with it. And that was the key: to figure out how to build the loop on this machine. The word processors of that day would lose characters when you hit a word wrap point. We couldn't do that. So that loop and how to get that was important. And of course we used Multics time-sharing time for it. We used the big computer to help us develop for this little computer. So that was in the end of 1978 and the beginning of 1979.

Campbell-Kelly: Really? You didn't actually develop the program on the Apple II itself?

Frankston: Well, yes; the Apple II didn't have sufficient tools.

Campbell-Kelly: So how did you develop it?

Frankston: Like I used to do. At ECD we had cross compilers and this was standard technique. Technically I had a program that – you wrote Assembler on Multics, super fast 120 character per second teletype…

Campbell-Kelly: Where did the Assembler come from?

Frankston: The Assembler on Multics was the one we used at ECD and it produced hexadecimal codes which I downloaded. I wrote a program on the Apple II, a terminal program, which downloaded the code and converted it to binary that could be executed.

Throughout this I was also writing tools. So I had to write a tool to act like the terminal program on Multics on its own . . . I forget whether I switched over the modem or how we did it, but we got the hex codes into the Apple II and then that became the executed program which we debugged.

Campbell-Kelly: To reiterate: you put the program into the Apple II, then you would play with it and get the feel of it . . .

Frankston: And debug it, yes.

Campbell-Kelly: And then you'd go back to Multics and redo the code and download again. How long did that take?

Frankston: The cycle? A while. Because 120 characters per second isn't very fast but then the machine was pretty small, so it was not a huge amount of code, but it would take a while. One of the big challenges was staying awake with your work at night. One of the things I learned earlier – time-sharing – when you did a long compile, you'd put your head down on the terminal and fall asleep and it would ring a bell or something to wake you up when it was done. So there were all these kind of techniques.

At the same time, by the way, we were also doing the business things. On Jan. 2, 1979, we aimed to incorporate and we met at a place I called Kentucky Fried Fish – it was a Kentucky Fried Chicken that had been converted to a Greek fried fish place. And I came up with the name Software Arts. It's significant that we chose the word Arts. I came up with the name, but it was important to choose the word Arts because it was engineering, but there was not this art versus engineering. Style in the substantive sense was important because… there's the old Charlie the Tuna ad: You want tuna that tastes good, not that looks good. But there was an aesthetic in the code; it had to be understandable, to make sense, to fit together.

So that theme of it being an art in an engineering sense was important, and later Electronic Arts picked up on this theme. As a matter of fact, they might have called themselves Soft Arts, but that was too close.

So it was always important to consider all these interactive elements. It's not that we were keeping all these different things in mind, but you have to remember, they had to fit together. So there was a synergy rather than a collection of disparate details that all went together. Yet it keeps going back to the theme I keep repeating: the program was a means of accomplishing the goal. We would get feedback – Dan would try it for the professors. We needed space to do enough financial planning; we need enough space to do this. And I tried as early as I could to do my taxes, that's why I did "lookup." What would you need for a checkbook?

So we were our own test market, because one of the other lessons that people still don't learn is you can't ask users questions for what you should do when they don't know the context. Products work best when you're using them yourself. One thing I like to do when working on a program is to use it before reading the manual. I want to give feedback while I'm dumb. Because once I understand the

program, I can't give any useful feedback. So you want to capture naive experience and discover your own mistakes. What would confuse me? And extreme laziness is a very important skill.

I mean, I can work hard to make it easy, so the net work might be harder, but I'd want to be entertained and make it interesting while I'm doing it. So you had to really keep all that in mind. But at the same time, there was a need for obsession. There were times, I know, I will find this bug. You might spend a week on an obscure thing and you had to have some sense – not the spirit "damn it, I will get this no matter what," it was almost the spirit of "I will get it." It's a subtle thing. It wasn't so much that I was going to climb this impossible mountain; it was: I will get to the other side of that mountain and I know I will do that.

Bricklin: I want to get to something Bob said a while ago and emphasize it. About figuring out how to make it fit and work on that machine because I would come up with something like the "help" system. I wanted to have this great help system where you could always hit the question mark and then the screen would change into showing you all the things you could type at the time and explaining it and stuff – this was all in the prototype.

The realities of getting things to fit in it – looking at my notes on the 11th of November, 1978, that's when we had the meeting with Dan Fylstra, I guess this must have been the one at Joyce Chen's. We wanted to be in 16K of memory. "Aim for ROM-able," it said – we wanted to be small enough to fit on a card because we didn't know as much about disks. "Less than 10 seconds for redoing the screen stuff." We were starting to have certain requirements and specifications that we aimed for. Aim for upgrades. Have menus. Home use, too. Because Dan wanted to make sure it was useable for home, too.

This is where the $35 price, $12.50 royalty, percent down, percentages down with dealer sales assuming $14.80 cost and I explained all that yesterday.

We then started renting typewriters and whatever. We had to rent a typewriter! And this is talking to the lawyer to do this. So basically, it's all the notes about that stuff. We were working with Personal Software at the time. I would take the photographs for them for their ads, because I knew how to take photographs and I could take better photographs. I knew about using a long lens and I have specs about what to do. Flatten it, use a shutter speed slower than a thirtieth of a second, etc.

But this thing of figuring out what to put in and what not was critical. Bob got to the point where – the text that was used in a prompt was also used as the value to test for when you pushed the button.

Remember Bob said about them using the names a certain way? First of all, we had to use A1 or something so that 1+A1 was distinguished from the 1+1A – that wouldn't work because you can't parse it as easily. It told you as you were typing what was happening. It wouldn't let you type in

anything illegal. It would on the spot make a beep. To make it easier to use we had features of that sort.

Its save format was the keystrokes necessary to recreate. So that ">A5:1+A4" (colon was synonymous with character return) would get you to A5 and put "1+A4" in that cell, so we could actually use that and it gave us the possibility of doing macros. We knew you could read a file in as a macro. We were just doing everything on top of each other to fit in the small amount of memory.

So, for example, we were trying to get it small and as it went on, unfortunately there was some pressure for us to have sine and cosine, but Bob didn't really get it in at first or something. He threw it in, but we found the book; they told us how to do it and stuff. But unfortunately, we had told Dan Fylstra's friend, Carl Helmers, that it was going to be in the product. And in the first mention of it in Byte magazine, the only example he talked about used sine and cosine. So we were stuck with making sure that sine/cosine worked and shipped and we couldn't cut that product feature out. Unfortunately, that lost 1-2K of memory, which was more than 10-20% of the available space.

So these were trade-offs in those days; remember, we were used to working with huge machines of that day, as well as tiny machines. So this was that interplay of what to throw out, which was going in. And the feature set was being decided at the end of 1978, with all those diagrams and stuff, and then the specifics of how to cut it down to be something useful and have all this neat stuff in a small space. It still does stuff that Excel barely does, in terms of synchronized scrolling, the multiple windows with different sizes at the same time and all that. Trying to fit it into 16K – we ended up at 32K. That was what was going on.

Frankston: Again, we could spend all the time on the little decisions. Not only did we save the keystrokes to disk file – it was in reverse order. Because we wanted to allocate the lower right hand corner to so that you didn't have to do a space reallocation while you're loading it. The cassette version as designed left spaces on the cassette, so as you read it in you could do the calculations and because you couldn't stop the tape. There were all these kinds of very low level details, but it was always against the larger sense of: how do you keep the experience going.

Bricklin: One quick thing before we finish with that. On the 7th of March, 1979, I showed it to my current finance professor, Professor Glauber, who has gone on to do a lot of things for the country in finance. I've written down here: " #1 – NPV, end of years for each. IRR – the Newton R search; don't think people would do it by hand. #2 – growth would be nice. Min, max, single step." He's telling me the features that should be in it, which was kind of cool, but NPV and the end of year for each – I guess that's why we ended up doing it the way that it was because it came from one of my professors.

Frankston: One other thing we did was decimal arithmetic. All the spreadsheets do binary now. We did decimal because we wanted to make the mistakes the same as an accountant would do by hand. And we also had huge precision – I think it was 12 [decimal] digit precision, which became very useful in some of the later applications like doing the U.S. national budget!

Bricklin: We had to figure out how big a company would use it, and would they do it in dollars or pennies. These were huge decisions because it changed memory and the way memory was allocated. It could completely change how much memory a single problem would take up. As Bob said, the reason I really pushed for decimal, which is the same as the calculators, is so that you don't want to end up with .999 that you get from binary. I wanted regular people to feel comfortable with this thing.

Frankston: Yes, and this was a case of over-engineering in some cases. We did it better than it needed to be.

Campbell-Kelly: Tell me about the user reception of VisiCalc.

Bricklin: You can get some of it from Dan Fylstra. Some of it we got second-hand. He would show it to a few people. Apparently Apple liked it, Markkula [Apple CEO], and Atari liked it is what he told us. And he showed it to Ben Rosen. He ended up showing it semi-publicly or privately at Ben Rosen's conference right after we signed our agreement with him. That was the first time it was shown semi-publicly and Ben's reaction to it is public as he wrote it up in his newsletter.

We then showed it privately to dealers and the press at the West Coast Computer Faire, and that was relatively positive. That was in May 1979. The public announcement and demonstration was at the National Computer Conference in New York City in June 1979, right before I graduated from Harvard and right around Bob's birthday. So we all went out for Bob's birthday.

At that point we had a little desk, like a table. In those days the PC business was considered the orphan child, the hated step-child of the National Computer Conference which was the big iron, big computer business. We were at the Hilton while the big thing was elsewhere.

Frankston: Was it in the Hilton basement? No, it was in the basement of the Sheraton Americana, which is now the Sheraton.

Bricklin: Basement, it was horrible. However, I've heard since that there are people who walked by there who were completely blown away by it, who then evangelized. They found out and asked how could they possibly buy this thing and then evangelized it with the companies. And the evangelizing of it by individuals I think is extremely important to how the thing was accepted.

However, it wasn't picked up by the press, other than in the New York Times where there was a funny article by Francis X. Clines, where he talks about the National Computer Conference and he says that Racal Vadic sounded like a gag reflex of the brain. It was "a layman's trip in the mega mega land of computing."

Each of the booths had signs and stuff and he happened to be there when they were writing the VisiCalc sign. And he wrote, "Even as the faithful gather and the sign painters are added to the Pantheon writing VisiCalc in yellow and black letters," or something like that. "VisiCalc, all hail VisiCalc." That was the only mention in the public press, in the non-PC press. The mainframe world dismissed us completely and so did many in the PC world, because we were business oriented and not hobbyist oriented. Unless they were business people, they dismissed us. We would show the product to people and the reaction was different. I found that the people we would show it to – if they were computer people they said, "Well, I can write a thing like that in Basic, you know? What's so special about it?"

Non-computer people would say, "Well, computers can do everything. What's so special about it." But if you showed it to an accountant, or somebody financial – if you showed them their application that they were doing by hand, they would start quaking and they would get so excited about it. Of course, salesmen eventually learned to sell it that way.

At the NCC, Bob wrote a scholarly paper about VisiCalc, an incredible paper, which he'll tell you about. When he delivered it, the interest was such that – of the people in the audience, there were only two, I believe, who were not related to either Personal Software or us, and they walked out. Bob will tell you that story.

Frankston: Actually, that's one thing we'll ask the museum for. I'm having trouble finding a copy of that. It was the 1979 SJCC (Spring Joint Computer Conference) where I gave the paper. I basically used examples of the telephone system in which I pointed out that in the 1930's there was a study saying that by the 1950s everybody would have to be a phone operator for the phone system to work. And by the 1950s, indeed, everybody was a phone operator, by making a dial and making it easy. And VisiCalc made everybody a programmer. The people did come to the more interesting talk, the undocumented op codes for the TI59 calculator, and they didn't stay [for the VisiCalc talk].

As I pointed out, everybody in the country could be a programmer, and the only person outside of the friends and family, it turns out, was Whit Diffie and he forgot that he'd been there. He was there.

It's interesting, I viewed being in the basement as not being officially part of the show, but it didn't feel like second class in the same sense, and maybe because I was more involved in the mainframe world – I was used to the SJCC. As a matter of fact, two years later I was on the program

committee for what became the National Computer Conference before it disappeared. And I remember attending the 1965 IFIPS Conference which was one of the predecessors.

In those days it was this great thing – I'd pick up all the literature. I remember one presentation where the slide started to melt as it was being projected so you had to go very fast. But by the time I was running the NCC I realized I wasn't attending the talks, I was too involved in doing and creating things.

Campbell-Kelly: Was your paper published?

Frankston: Yes, in the Proceedings.

Bricklin: What was interesting – Bob talked about the Procrustean Beds of the programs that you buy off the shelf, but how VisiCalc lets you tailor it to your own needs, which is this whole concept of people being able to do the programming themselves, much like they could be your own telephone operator. I had never heard of the term Procrustean Beds until I heard the story, I guess. And there you're using those high-fallutin' terms of how you wrote that. Bob's a really good writer – it was a really good paper.

Frankston: But when you look back, as ancient history, and it's easy to forget that we weren't that primitive. We actually weren't using sticks to make fires or something. We used online services; we used communications networks and all these capabilities. For example, one of the design decisions in VisiCalc was what NOT to use. As I said, we used to display the graphics also used by Macsyma – symbolic representations. We decided to do concrete implementations for symbolic was a conscious decision to match user models. For example, not use precedence in the equations. It was an explicit decision to match what we thought humans did – we didn't realize people who used calculators actually WERE used to precedence. So there was a lot of awareness of the abstractions and the concepts.

Bricklin: I think this is an important issue. While there were many hobbyists in the industry who were at the hobbyist level who cared about the hidden op codes of the TI calculator, there were many of us whose background was very strongly in mainstream computing with formal training, experience on all sorts of machines. We're not talking about just me and Bob, but we were an example of that, but even Gates and people like that who'd been involved and had worked with all sorts of machines.

Yet the other parts of the computing industry didn't pay attention to us much, which is why you'll find different contracts and stuff like that. We didn't share with them. We felt like step-children to the wedding until eventually we got to the point where revenues were such that Apple became a big deal – they had to accept us. That took a few years. If you look in the popular press, other than Apple ads, the concept of using it for mainstream applications – it was not accepted. VisiCalc was

never really a "Guess what! VisiCalc is here!" article. Like now – when a computer thing is announced, way before it's shipped it's hyped all over the place.

But you didn't see it then. In fact, in some magazines, other than maybe a mention in an article about something else . . . I think in Forbes, it wasn't until they listed what computers to buy, it had a column "VisiCalc or Not," there wasn't much [coverage] . . . remember VisiCalc was announced in June 1979, available for the press to see . . . until like in later 1980 or 1981. We're talking about – here's something in hindsight. So that is part of this trying to figure it out. But there were many people who got exposed to it who got it – more than we got for their area – and evangelized it.

Frankston: Ben Rosen.

Bricklin: Ben Rosen was one, along with Al Sneider and the other people who bought the first copies of it. Mort Rosenthal begged Personal Software to get the first handmade copies that we had that these guys got to be able to use, where we hand did everything. So there were those . . . . and they were evangelizing it out there. If you look at the actual sales, they were pretty meager, if you think about it. We're talking about them averaging 1,000 units a month until Radio Shack started advertising it; Hewlett Packard started advertising it when they started having their own versions.

Frankston: It's interesting to see Dan's perception about being a stepchild. One of the experiences I had in computers in the 1960s was this experience of knowing more than the professors. We were professionals; the professors were just learning – they were recycled EE's in many cases. Some, like Corby, Jerry Saltzer, and Peter, had, of course, been in the field. But for the most part it was a new field. I didn't even start out thinking I was going to go into computers in college because it wasn't a real field – it was too easy. It was a hobby. And when I decided to jump off the mainframe ship and get onto personal computers, it was freedom for me. I mean, this was hard to explain to the mainframe people. Yes, I was giving up all these miles and miles of books and all these protocols, but here I could revel – I could do what I want! I'd been used to doing system programming, like working with David Reed – he modified the lowest levels guts of Multics at Interactive Data. Back at White Weld, I was working the operating system, lower levels.

The idea of being limited to just writing the application program and not being able to modify the terminal drivers – well, that's no fun. You wanted to be able to do everything. I remember one of the things about the Multics terminal – they tuned it so the typeball going a certain speed and you hit it at the right spot.

I mean, all those . . . there was a fun and a freedom aspect to this – that we could do whatever we wanted. You can speak to Wozniak and Gates about this…all these design decisions – there was this ability to experiment and I think one of the frustrations of computing now is the extent to which we've layered so much stuff on that you don't have the freedom. There's a positive side – you get

good leverage. In other words, I like getting the tools that give you leverage, but at the same time I also know . . . I remember there was a demonstration at one of the Agenda Conferences of the Be computer and the BeOS. People were excited about how fast the graphics were. Then I realized – these people had never seen an unencumbered machine. There was that aspect of being able to re-think everything that was very important and it's easy for people to forget that.

Bricklin: You can see how there was this difference in terms of us being put where we were put and the interest that was shown. It took a while, but eventually it got accepted. There were evangelists in every company and those little evangelists started getting things going.

Frankston: One other thing about the field in those days – to get a sense. As I said, we were on the Multics project. There was a group – I'm not sure the details – but they were impatient for Multics because they wanted it to run the phone network. So instead they worked with IBM on an even more ill-fated project which was TSS. But a group of them kept enough hands on Multics and they were the ones who programmed Multics, and they said they created a personal version of Multics. They called it Unix – which is a nice play upon words for a castrated Multics.

The original Unix was a bare veneer to like Multics, underneath it was not Multics. But it had the important point of being cheap. So they jumped early in the 1970s into the small cheap computers and showed that price is an important factor of what a product is. Eventually that's grown up now to be a big system, but I remember in the early days when we were on the Apple II, it was hard to explain to the Unix people why the constraints of time sharing aspects were actually in the way of reinvention.

Campbell-Kelly: Once VisiCalc got into the field, how did you support users – were there help lines?

Bricklin: First of all, things were sold in shrink wrap – you bought it and went off and did your stuff. We were the developers, so any question about interaction with the customers all has to do with talking with the people who were marketing it and selling it. The users wouldn't talk to us specifically.

It wasn't a very complicated thing to use. We did produce our own publication to teach people. We came out with a product called SATN (Software Arts Technical Notes), which we can get into because that was a bone of contention between Personal Software and us. They threatened to sue us over it. There were all sorts of things – that gets into the relationship between the two companies.

But remember the way it was – products came out with a manual. In the case of VisiCalc, it came out with a reference card which had been tested and found to be good. I found in my notes that I would bring classmates in and give them the reference card and the manual that I wrote to see if they could figure out the product. I learned that people didn't read it, they just started using it. I

made sure it was that way. Most people learned it by looking at the reference card, maybe reading the manual or learning from somebody else. What was there for us to teach them?

Frankston: I can't over-emphasize that because the mainframe background – they put a system engineer in every box – you need a large cabinet so they can live there. OK, affect doesn't carry in text, obviously I'm kidding – but mainframe users paid a lot for support. For a consumer product you had to make the product obvious. If you called for help, we had failed – it was too complicated.

It was important to recognize your audience. VisiCalc sold to people who appreciated it. Ordinary mortals just didn't care or didn't get it. Many didn't even know the program did calculations so they put calculators on keyboards. For many the value was simply selling ideas, the CYA aspect – cover your ass – as a presentation tool for some. The main thing is – the market for people that appreciated it was large enough to create the illusion it was for everybody, and that was reflected because the people who used it were the people who invested in it. So you get these factors.

Bricklin: It was expensive. While VisiCalc originally cost $100, they kept raising the price. Ed Esber mentioned to me once that the more he raised the price, the more were sold. It turned out that's because they were dependent on the program. People generally walked in to buy a VisiCalc machine. They had to buy the machine; they'd buy a good printer – they might buy a Diablo printer for it. So they might be paying $5,000 for a machine to run this $100 to $150 product.

This was not something for casual use. This was used by people where it mattered to their business, which was good because a lot of them were money people – they were investment people and all, and they saw that the personal computer industry was now valuable and they put a lot of investment interest into that area. So that was a helpful thing. Accountants – it made accounting fun. It was used to negotiate mergers and sales and all. It was used to do anesthesiology calculations during open heart surgery – all sorts of things. So these were people who could afford it. These were early adopters, and it was a simple product.

Frankston: But I think it was also important that we were naive enough to underprice it. I think that set a precedent for giving it the illusion that software was for everybody. So I think – as much as maybe we could have charged $10,000 for the machine, we created the sense of it being a large marketplace and also we set sort of a price ceiling on software, so you can get multiple applications. People might spend $2,000 but you had to spend another $2,000 on word processors. And as the machine cost came down, we are caught in a greed problem where the software costs more than the machine. So a lot of this turned out to help create the industry.

Bricklin: In terms of the classical maintenance, improving the program and stuff, that was in the contract. It was assumed we would be improving the product. But there was this large

number of computers. At this time in the industry, there were all these different proliferations of computers. So you had to produce the same product for as many computers as possible. That, and keeping it the same for maintenance purposes and all – but for maintenance purposes and for keeping the documentation and not having to rewrite and all that – and training and all – all that stuff together, we sort of kept it the same and then we would lock step and have to come up with upgrades for all the machines, and at the same time come up with new versions of it. So we were working on VisiCalc II, eventually called VisiCalc Advanced Version which we had. There was an intermediate one – Apple III had one with special features in it.

In those days you were running in a million directions at once. Later on, it was found out in the industry, you could rifle shot one platform, the IBM PC compatible platform and that was sufficient. And you could use different technology doing that than the technology necessary to do all platforms. But at that day, if you look at the model that Personal Software had, they had the PET; they had the TRS-80 and they had the Apple. Two different processors, three different types of machines, different ways of selling it.

A lot of it was sold directly through manufacturers; the Tandy stuff was all sold directly by Tandy. Why did a lot of VisiCalc sell near the end when Lotus 1-2-3 was coming? 1-2-3 came out in early 1983. It was selling incredible quantities over that year – we were in the midst of our lawsuits and stuff – so by the time we started selling our new version, 1-2-3 had sold over $100 million – it eclipsed VisiCalc sales.

Why was VisiCalc still selling? IBM was still buying large quantities of it because they had a big company and decided they would place an order for this many under their name and there were things like that to keep it going, there was hysteresis in the system. You sold through the manufacturers. And you had to do deals – all that complicated it, until it ended up being that you could sell individual products through the dealers, eventually using distributors, and then eventually bundle them on the machine. They kept changing the way you could do it, where you could narrow and narrow it down, that let us put more into development. Next question?

Campbell-Kelly: I think that's about it.

Bricklin: I thought we had till 2:00.

Frankston: There's a whole separate thing on the business aspect. Then there are the next 20 years! I'll send you a copy – remind me to send you an annotated version. *[Here there is a total context switch in reaction to Dan Fylstra's essay with his assertions about what happened after VisiCorp sued Software Arts. —Dan's editing.]*

Bricklin: This whole thing about VisiCalc being killed by us marketing it through the mail is bizarre!

Frankston: They stopped advertising it the year before.

Bricklin: Personal Software only spent like $10,000 advertising it the year before – when the competition was spending much more. Microsoft was spending millions of dollars advertising. VisiCorp was saving all their money for their VisiOn stuff. They had lost interest. They had people – they still had to sell it. They were still selling a lot of it on the TRS-80, the Apple II, the Apple III, so they still had revenues coming from other versions, but the IBM version, in corporate sales, all had switched to 1-2-3. That was gone.

Frankston: Again, they threatened, if we added feature stuff and kept it up to date, they were going to sue us.

Bricklin: There was a whole lot of things leading up to it, but to think that the reason that 1-2-3 succeeded was that in February or March of 1984, a year and a quarter after 1-2-3 shipped – a year and a half after it was announced, that we sold VisiCalc partially to dealers, but partially direct at a lower price. To think that that was what killed it . . . we had that low price on it because 1-2-3 was winning and we were trying to compete!

As I recall the time – they [VisiCorp] were completely aiming that 1-2-3 was coming on, and they were going to take it on with VisiOn – Quasar, as he said and VisiOn Calc. We kept on waiting for them to ask us, formally, to develop for it, but they never did. That's a contract thing. They did not seem to get it. There was this blind spot to 1-2-3. I brought that up to Dan [Fylstra] and I said, "You've got to look at what Mitch [Kapor] is doing." And Dan said to me, "Mitch is a nobody." There were relationship problems between them.

Frankston: Remember, Mitch was their creation. They pissed him off and they dared him to do it.

Bricklin: Mitch was upset by the lowering of the royalty rate. Mitch wanted to do his own company and whatever – and what type of person he is, that they can talk about, whatever. Mitch has treated us pretty well throughout, considering he was a competitor. But they [VisiCorp] didn't seem to take it seriously – there had been many types of competition. They succeeded against many types of competition. And they were so banking on VisiOn – how it was going to be . . . and still calling it disruptive. It was not disruptive in the terms of that. You can get into the technical reasons why VisiOn didn't make it besides the fact that, unfortunately for them, the hardware wasn't able to support it well.

I believe there were some user interface problems with it. But that 1-2-3 came in; that the dealer channel they lost early to 1-2-3, and we're talking about a year before we had this stuff. It's disingenuous for him to say otherwise.

Frankston: I was also going to mention one mistake we made by the way. Mitch actually wanted to join us – we should have done it.

Bricklin: We should have done it, we should have hired Mitch. Mitch wanted to work for us and we didn't need a product manager at the time.

Bricklin & Frankston Interview – Part 2

Paul Ceruzzi: We're continuing the interview with Dan Bricklin. My name is Paul Ceruzzi. I'm taking over for Martin Campbell-Kelly. And we're going to pick up where we left off. [Bob Frankston joins the interview a few minutes later.]

USE OF PRIME FOR DEVELOPMENT

You were at a point where you were discussing the question of getting a time-sharing system to do development of VisiCalc. So there was a Prime computer used for development work?

Dan Bricklin: We borrowed with personal guarantees, Bob and I, the money to buy a Prime 350 minicomputer on which to develop the product. I borrowed money from my folks, I think. Bob got his money maybe from his folks, and we personally guaranteed the loans. And that's how we bought this Prime minicomputer.

Ceruzzi: For the purpose of developing what?

Bricklin: VisiCalc. Bob then wrote an assembler and a linker. I wrote an editor. We had to have an editor because they didn't have what we wanted that would produce cross compiled stuff for the 6502 and later for other machines. And that was the development environment that we developed in. To be able to pay the monthly payments, we had signed an agreement with Personal Software and they gave us royalty advances which had to be non-refundable because the bank wouldn't accept it otherwise. We never got venture capital in our company. I can give you some dates on how some of that stuff worked.

Ceruzzi: You saved your Harvard Coop notebooks.

Bricklin: Yes, my Harvard Coop notebooks and all this stuff. Let's start with the Prime Leasing, we could put up stock, the pledge, whatever. Securities. We went through all this type of stuff.

Ceruzzi: Did you have any discussion about Prime versus other vendors?

Bricklin: Yes. We looked at DEC and all. But the thing about Prime is it had a PL/I compiler, as I recall. Bob can get into details about that. It was very much like Multics. It was developed by a lot of ex-Multics people. So we were used to that. And Bob eventually ended up writing an Emacs for it, a full Emacs for it, which we sold back to Prime in return for some hardware and stuff, which we used as our mail system. Our mail system was really cool, which we used for everything.

Ray Ozzie liked it and it re-inspired him to go back and build Notes, at least he's written this, and inspired him to do Notes, remembering his experience with the Plato System.

Ceruzzi: So it goes back to Multics?

Bricklin: It goes back to Multics.

Ceruzzi: Multics was done on a Honeywell machine?

Bricklin: It was a GE machine later bought by Honeywell and stuff. So at this time that we got that system, we were back and forth. You talked to Dan Fylstra. We found a typo in the Bridge manual, you know.

Ceruzzi: Do you have a date on this notebook?

Bricklin: I have notebooks that start in late 1978 and go on talking about, which I've already looked at a little bit, the prototype of VisiCalc. I kept notes basically of a lot of conference calls, discussions with people about defining a demo, do they have to commit that early. Every page has stuff. This was in 5 April, 1979. We were interviewing accountants and were asking . . . Ah, here: "PL/I G not until late May, but with debugger, go to I Mode, should be faster." This was all about Prime. We went with Prime pretty much because they had the better languages.

Ceruzzi: They were a local company also.

Bricklin: I had worked at Prime the summer before and found out about it. But I used my contacts at DEC and Prime. We also looked at a DEC and at DG [Data General] with its AOS system. But we wanted to have our own system and we built our own system. We shared space. We got space from a friend of ours from the Multics days, shared space, a consulting area, a development and consulting firm of some sort.

Ceruzzi: That's where the computer was installed?

Bricklin: Yes. [Looking at notes:] April, 1979, we talked to John Strayhorn, maybe we could find some consulting work or something. We discussed sharing facilities.

Ceruzzi: Where was that located?

Bricklin: It was in the basement of a building at the corner of Massachusetts Avenue and Main Street. We have pictures we can show. [Reading notes:] "We got a 32K PET, sent option to a lawyer." We were still negotiating a contract. All sorts of stuff. You get the idea. We got the Prime. I don't remember when the Prime came. But eventually the Prime came and we wanted to

make sure that when we finally shipped VisiCalc, we would have our own code produced by our own compilers. And we finally had the test versions in the summer and then shipped it in the fall. It was all developed on that and we had the space which we shared.

Ceruzzi: For legal reasons?

Bricklin: Both. We couldn't afford to spend money on time-sharing. We could buy our own machine. In those days, development – a lot of people did development, as Bob explained, on another machine. And then from that other machine, you loaded to the micro because the development systems on the micros weren't up to it. That's how Microsoft Basic was done – that's how Microsoft ended up having a PDP-10 I think. They eventually used XENIX to do their development. We did the same thing. We developed all our own tools over the years. We improved the tools. We wrote our own implementation language, a higher level language. In fact, that's one of the issues that eventually came in, is that in the early days of the PC industry, there were so many different machines and you didn't know which was a winner. You sold to each manufacturer. So you had to port all over the place. That's what Digital Research was – a porting company. Microsoft was a porting company. That's what we did. We had to figure out ways to port the same product and cookie cutter it out. And everybody went a little different and you had to fight with them. Otherwise, the costs would go up. So we eventually moved things to a higher level language. We did our version 2 of VisiCalc, in a higher level language.

Ceruzzi: What language?

Bricklin: We wrote it in something we called IL, which is a Lisp derivative. It was like writing in Java or something like that today. An interpreter. Microsoft had a similar type of thing for Multiplan. They wrote in a language which let them use a cookie cutter to put it on many different machines. But the Apple II version of the VisiCalc II (VAV) was written in assembly code. We realized that to port that was going to be so expensive. When we ported from the Apple II and Apple III, doing the IIe was next, then to port to the IBM PC it's a different code base. The way we did the VisiCalc code base is, since we had our own tools, we hired Seth Steinberg, who had worked at the [MIT] Architectural Machine Group – Media Lab, real experience, real bright guy, helped bring a culture into our company. Free Cokes came from him. He ordered it and all that stuff. Lotus and others all copied this. It helped bring that type of environment from MIT into our company and spread hopefully to others. What Seth did is he said, "I'm going to do an idiomatic translation, basically, from the 6502 code to a Z80 code to do the TRS80." And what he did was he modified the compiler to list the two sources synced on labels. The compiler we used had macros and it had no 'go-to's in it. Basically it was all IF THEN ELSE and stuff. It was a macro assembler. He basically left the comments in the 6502 code and just wrote Z80 code that did the same type of stuff. We found out sometimes the bugs were actually the same. And when you printed it, the listing actually listed the comments from one to the other and you could compare the two. That showed immense maturity on his part to be able to take somebody else's code and as a coder, an

innovative person – he was incredibly innovative – to do that and to get it running on the Z. That's how we got it running on the Z80. His name was Seth Steinberg.

Ceruzzi: Did he stay with the company?

Bricklin: He was with the company until much later when we started downsizing or demise, whatever. He is currently living in the Seattle area, I understand. But Seth was responsible for lots of stuff. So that's how we took advantage of the big machine, had tools like that. When the IBM PC came out, we needed to convert the Z80 code, and we eventually moved to a CP/M version as we had done for HP. We moved that to the 8088 where we modified our assembler. My cousin David Levin modified the assembler so that it would do whatever could be automatic and flag the things that couldn't be done right to aid [us], so that when Jeff Stevenson, who is now at Microsoft and who has been other places, Jeff Stevenson did the actual conversion. He was able to get the IBM PC version up really quickly, based on this code that worked. From IBM's viewpoint, we were told, we had like no bugs. We tested well. The whole thing in those days was working off of this. That made it hard to move to new versions. It made it really hard. But we did come out with these new versions. We did try to switch to a higher level language to do that. But in hindsight, that ended up being slower.

DELIVERING VISICALC

Ceruzzi: Except that once you got on the PC, you didn't have to do any more porting.

Bricklin: But we didn't know that at the time. You have to remember, that's in hindsight. So it was launched in 1979 for $100. Why did you decide to raise the price? Everything about pricing was decided by the marketers. You can ask VisiCorp about that. Manufacturing process, ask VisiCorp about that. What did we physically supply to Personal Software? Initially we supplied them with diskettes, which were copyrighted with the copy protection already on it that we had worked out together, the copy protection for the product.

Ceruzzi: How did that work?

Bricklin: I don't remember the details of it. Bob Frankston would know. And Peter Jennings. For different machines, we used different types of copy protection. Which wasn't very good to do. And whole companies were built around making VisiCalc copies. One of them eventually became MacAfee. But we used to supply these master disks and there was a lot of discussion about when we would okay the manufacturing. Because we didn't want to let it go until it was okayed by the QA people. We provided diskettes that were master diskettes and then they produced from the master once it was okayed. Later on, and this was – we kept having little problems – one day the Supreme Court in California made some tax decision where you couldn't send master copies. It had

to be non-tangible or something. So we suddenly had to send it electronically. On short notice. But it had to ship that day. So luckily, we had our machine that we were able to set up in a way where they could dial in and we told them how to be able to do that. So from then on, all the shipping was done electronically.

Ceruzzi: It was a relatively small number of bytes.

Bricklin: Well, not at 300 baud. A question, did I ever meet the early users? First of all, we were many of the early users and yes, I was constantly bringing in test people and showing people. Once they actually got out there, Personal Software would try to keep us as the developers and they were the marketers and try to keep a wall between the two. But obviously I ran into a lot of the users of the product. I remember showing it once to The Computer Store, which ended up being one of the major vendors – The Computer Store here, from which the main selling people ended up being principals at Lotus later on. Mark Goldschmidt and I can't remember his name Rosen – not Ben – he ended up being head of sales. But also Joel Skolnick who was their financial person. "This is kinda cool." And Joel started shaking. And he said, "Lemme show you. Quick, quick, quick. Look, I can do this! This is great! I do this all day, takes me a week. You can do it in no time, like 15 minutes." I got the idea then that we had a hot product.

[Reading from questions:] "VisiCalc had many imitators such as SuperCalc and Multiplan." They were clones, especially in light of IP protection. We always viewed it in those days that that was progression. The people built on the ideas of others. Just don't copy our code. And if somebody pirated the disk, cloned the disk and did it, we sent people after that. But if you're writing it from scratch and you're making a better product, that's our problem. We felt bad about it, and we hoped that our marketing people would be able to go against it. But if we could have, we would have tried to get a patent. We had discussions about patenting but weren't able to get that. Unfortunately. We brought a patent lawyer in and told us it wasn't patentable. As Fylstra related, the patent lawyer said the chances were slim. Because that was before the days of software patents. So to get a software patent, you sort of had to sneak it in by making a machine. And the chances were slim and it cost a lot of money. So the decision was up to Personal Software and they decided not to do it. But we were really sticklers – we liked our stuff – we were brash young kids.

Ceruzzi: But you also felt like you could compete.

Bricklin: We had great ideas for new versions. And the VisiCalc Advanced Version had variable column width, it had macros, it had multiple sheets. We supported multiple worksheets simultaneously and 3-dimensional, and the whole bit. But we shipped it on the Apple III. That was a problem in terms of that, of being able to do it. We did want to enhance it. It was a problem in terms of being able to take on the competition.

Ceruzzi: The problem was that you shipped it on the wrong computer.

Bricklin: Wrong computer, plus we were too busy cloning. Plus at that point, the relationship between the companies had deteriorated. I should get into some of that. I think that because that relates to the whole reason of how VisiCalc was enhanced or not enhanced and stuff. And we get Mitch Kapor a little bit. Mitch was a person that we knew locally and was a friend of ours. At one meeting – he was also a software developer – and he had this thing called Tiny Troll which he was going to develop as VisiTrend/VisiPlot for Personal Software. He was an author. Fylstra was in town once on a press tour, I got Mitch to come to dinner with us because Mitch was trying to reach him and talk to him. Dan Fylstra was saying he was having such problems hiring. So hard to get a product manager. And Mitch said, "What about me?" Next thing we know, Mitch is moving over there. [He is asked about dates.] If I have dates, I'll get them. That particular meeting, let's see what I have for that. [Looking at the notes:] I brought Dan Fylstra and one of his people, Skip Vaccarello, in to meet with Professor Porter, Michael Porter, over at Harvard Business School, who gave him a lot of advice. The second meeting was March 8, 1979.

Ceruzzi: I'm not supposed to interject my own personal history, but I remember Mitch as a DJ at WYBC at Yale.

Bricklin: What was Don's last name? There was somebody who knew Jon Sachs, who also went to school. It was a very small world. So on 31 March, 1979, I talked to Dan. He was in San Francisco. I guess they had already been planning to move. We understood why they wanted to move to California. But it was a real shame to have them far away.

RELATIONS WITH PERSONAL SOFTWARE

Ceruzzi: So the question is, what was behind the decision?

Bricklin: Well, it was clear at the time. And if you even hear Dan speak now, he was enamored by Apple, because they were doing everything right. And whatever happened, he got the same venture capital. The stories he tells about it were real good. I can see the reason why. As we've learned later, with Apple, it's much better to be nearby. And that's where it was all happening.

Ceruzzi: Did you have a personal relationship with Steve Jobs?

Bricklin: No. Only when they would come to Boston. All the relationships with the manufacturers were done by them. They were the publisher, we were the developers. We had enough things to do, plus they wanted to keep it that way.

Ceruzzi: So you never had an experience of Jobs coming and telling you what he wanted.

Bricklin: He yelled at us for why didn't we just stay on the Apple, why do we do versions for others. I don't know. His relationship was more later with Bob Frankston when Bob was at Lotus.

Ceruzzi: I don't know if this was covered in the previous interview, but did you have a relationship with Ken Olson?

Bricklin: No, I didn't. I didn't actually meet Ken Olson and have discussions with him until the Computer Museum was created and the book *Wizards and their Wonders* [ACM Press 1997] was published. I had seen Ken and heard him and maybe even asked a question. I asked a question once I think about personal computers at a shareholder meeting early on. When I first found out about the PDT. DEC actually had a Microsoft Basic machine. One of their machines had Microsoft Basic in it. The 11/05 or something. This was in the spring of 1978. Originally VisiCalc was supposed to be on a DEC machine, but I wasn't able to get one from the DEC salesman. And we had been negotiating up to the last minute trying to get one, to buy one, and get a PDT, and what to run on it. It would have been written in Pascal or something like that.

Ceruzzi: Would that have been hard to do technically?

Bricklin: I don't know. I don't know what decisions would have been made based on that. But I got to play with an Apple II. The Apple II had disk drives. We made the deal with Fylstra. Let's do that first and the other later. So DEC lost out from that viewpoint. Here [consulting notes:] "31 March, 1979 – talked to Dan in San Francisco. What about rooms for show? We'll do it where he is. See about traveling." This was for the West Coast Computer Faire coming up. "I asked about Ben Rosen. Ben Rosen said it's great. Regis McKenna thinks it's going to do well. I don't know how to spell it. Say – different price is okay. Don't want to confuse with small business. Want personal tool." Here it says, they want a personal tool. They don't want to be confused with a small business tool. This doesn't correspond. Now, one of the things we have here is a document from Dan [Fylstra] where he writes about what he thought, his view about what happened with the VisiCalc's story.

Ceruzzi: Is this the e-mail that he sent around? *[Fylstra circulated an essay at the 2004 meeting in Needham, but the document was not then publicly available.[1]]*

Bricklin: Yes. "The creation and destruction of VisiCalc," which is great, because this is his thinking, the best he remembers it.

Ceruzzi: That's what we're here for, to get people thinking about these things.

---

[1] A version of Fylstra's document, "The Creation and Destruction of VisiCalc," is posted at <www.edesber.com/companies/visicorp/history.shtml> (accessed July 2011).

Bricklin: But he does mention some stuff here and so I figured we would see what we can find out if any of this stuff . . . [looking at the printout of the email:] "Leading up to VisiCalc" – page 3 of the e-mail. "I saw great potential from the first conversation." Bob went into the last thing about this. He was looking for business, but he also was looking for personal use. As an MBA, he and Peter Jennings got it. They knew how valuable it was. But for some reason, I remember that fall, they were still looking for a checkbook program. The word kept coming up. And so we had to always show we could do that stuff. Now he does not think this is true, but here he even says they want a personal – underlined – personal tool. The image we had on the cover, I think he got to choose that. I'm not sure of it. One of the reasons is he says here that they made a good demo. Doing a checkbook. It was not a good checkbook. That wasn't its strength. But we had to make sure it could sort of do that.

Ceruzzi: That's what everybody thought in their heads that he sees what we're going to do.

Bricklin: And in hindsight, it's true with Quicken. Once they figured out how to do it. But the reason I say that is there was a reason they were asking about personal finance. These were personal types of things. Also he believes – I had actually worked with a personal computer. I had not worked with an Apple II. I had touched the Tandy and actually typed on it and got the feel. I knew what it was. I hadn't read the manual. But I knew all sorts of things in this class. Smaller machines and bigger machines at that point.

Ceruzzi: What would be an example?

Bricklin: I had worked with electronic cash registers. They were 6800 based. 16K. I wrote the DEC word processing system that was on a 16K 12-bit word machine. PDP8/A, on a desk, which was the "Classic." That machine was designed by David Ahl, who went on to do Creative Computing. [From the printout:] "Bricklin borrowed the first Apple II, took it home and never returned it" or something like that – according to my notes, I had to go to his place to use it and I wrote it there. Eventually, he had to lend one to us because it was always in our contract that they had to give us at least one of the machines on which to develop. Otherwise, we couldn't get it working. It wasn't like poor Bill Gates who did not have a machine, wrote his Basic without ever having tested it on the machine. We didn't want to have to [do that]. We knew we had to test on the real machine.

It says "Should we work for them or not?" Page 4. Bob and I wanted to have our own company, but later on we did talk about merging. One of the things here is a discussion of the contract at one point. "My lawyer says best efforts, matter of degrees, sterner test." He liked us using the words "best efforts" in the marketing. It turned out in a lawsuit later on, it was the fact that they weren't using best efforts in the marketing. They cared about VisiOn and they stopped marketing VisiCalc. That's why they lost the lawsuit. All we had to prove was that they had breached the contract

because they weren't doing best efforts. And then the trademark reverted to us. It had nothing to do with who came up with the name VisiCalc. There was this whole discussion about who came up with the name VisiCalc. Dan Fylstra told me he came up with the name. I believed it and said it for many years.

Ceruzzi: OK, let's get this Vics Egg on One [cafe in Cambridge, Mass.]. Is that place still around?

Bricklin: No, it's gone. Mass Ave. But I have a picture of it.

Ceruzzi: On the other side of Harvard Square?

Bricklin: That's right. Between Porter and Harvard Square on the left as you're going toward Porter.

Ceruzzi: This is important to me.

Bricklin: There's a photograph of it. They both agree, that's where it came up.

Ceruzzi: It kind of floated in from looking at…?

Bricklin: Whatever. First of all, I had always said Dan decided on VisiCalc, that's for sure. And we all agree with that. They both said they'd made it up. It doesn't matter. I don't think it matters. However, an example of why it would be, is that I told everybody including [InfoWorld editor] Stewart Alsop that Dan came up with it. There's a difference between coming up with it and deciding on it. But it wasn't until years later and Bob heard me speak when I said Dan and Bob said, you know that's wrong. So that's why in general you'll see it indicated, including in Stewart Alsop's article in 1981.[2]

Ceruzzi: We're being joined by Bob Frankston here.

Bob Frankston: I could send you an annotated set of comments.

Bricklin: It wasn't until years later that Bob said, by the way, I heard what you said, but I'm the one who came up with it.

Ceruzzi: Can we talk about the etymology of the word spreadsheet? Where did that come from?

---

[2] See Stewart Alsop II, "How Visicalc Works," Inc. (November 1981) <www.inc.com/magazine/19811101/8338.html> and "Software Arts Wrote The First Best-seller," Inc. (January 1982) <www.inc.com/magazine/19820101/5230.html> (both accessed July 2011).

Bricklin: It wasn't called that. It was called a visible calculator. It wasn't called spreadsheet by us.

Ceruzzi: So where did that word come from?

Frankston: Because it is a spreadsheet. It's like where did the word "table" come from. It was a generic term.

Bricklin: It produced spreadsheets. We were not involved with that. You can look at the terms we used by looking at the terms in the documentation that I wrote. That will tell you the terms that I used at the time. And what the terms became later came off of documentation written when people wrote about it. We called it "the sheet."

Frankston: I called it SS, spreadsheet, from the beginning. I have an annotated version I can get from the car.

Bricklin: [Continuing to review his notes:] It's like 18th April, 1979, we were still discussing "distribution. 3A, warranty, best efforts, delivery time, they request – they should take the risk." So we were back and forth with this contract until we eventually got it. What did happen is between the 14th and 16th of March in 1980, I was asked to go to the West Coast Computer Faire because Adam Osborne – I wasn't supposed to know – but he was giving me and Bob this award. The White Elephant award. Great things, a tape of the award is on my website.

Ceruzzi: What does White Elephant stand for?

Bricklin: I explain on the website. You can take a look at my website for that. We have to compress this, unfortunately. It was great. But Bob couldn't go because he was getting married that Sunday. So I took the red eye that night. I almost missed his wedding, but I got there with an hour to spare. But it says here, "West Coast Computer Faire, got Osborne's award for the most significant contribution to the microcomputer industry in 1979. Talk with Dan F., interested in us being new product/technical arm of PSI – merger." That's Personal Software Inc. So this was discussion of a merger starting in March of 1980. A few days later – throughout all this stuff, we were talking to all sorts of manufacturers, getting manuals on different computers – Wang – we talked to Wang. This is the 19th of March. I called up Professor Cash who was my professor, Jim Cash, and told him we're going to need an advisor, an independent board member because we now have to do this merger thing. Can somebody help us? He said he would explore it. The 20th of March, he suggested Julian Lange as a financial guy or maybe this other professor I had had. "He'll approach him, told me how much, advisory board. And set up a meeting." That was the day that Mitch started with Personal Software as a product manager. And he was our product manager, which was great. He kept East Coast time. We had known Mitch before.

Frankston: I demo'd VisiCalc to him when he was [unintelligible] librarian.

Bricklin: He was involved in the PC world early, early. Ed Esber was there in Personal Software. "Going to Apple to discuss the VC contract."

Ceruzzi: VC venture capital? [No, VisiCalc, responds Bricklin] When you went to Apple, who did you talk to?

Bricklin: I didn't go to Apple. Everything with the manufacturers except when the manufacturers insisted, which was only with IBM – because they had us as party to the deal because we were the owners of the source code, copyright – all of that was through Personal Software. They controlled the relationship. We controlled the technical. We had to split between that, which was fine.

Ceruzzi: So you were okay with that?

Bricklin: Always okay. In terms of that, all we cared about was we wanted to do all the products in the future. Make sure people knew that we were the ones that did VisiCalc.

Ceruzzi: People must have known that because you got an award.

Bricklin: Nobody knew. Who covered this stuff?

Frankston: Everybody was on the west coast. Especially when it became VisiCorp.

Bricklin: The industry was not covered like it is today. Some press did. We ended up getting a PR firm to help us. To do it, you had to do some work. Because as Dan says, they like to make it simple. Whoever they talked to, they liked to have the one, so they mention Fylstra. If you looked at what Stewart Alsop wrote in Inc. Magazine, he said the split was an issue that got people confused.

Frankston: And also remember the name Dan.

Bricklin: Dan confused people because there was Dan Fylstra and Dan Bricklin.

Frankston: So Dan did VisiCalc.

Bricklin: Yes, I was Bob Bricklin or something.

Frankston: But I'm saying it's one of these things. Sometimes linguistic things like that make a big difference. We've yet to recover from Christopher Columbus's little mistake.

MERGER DISCUSSIONS

Bricklin: 2nd of April, 1980. Dan – we're to have a meeting with him. A meeting on the 11th "goals: where do we want to be long-term, keep employees, get projections." Basically, we're starting to talk about a real merger. And we had to now value ourselves and stuff like that.

Ceruzzi: So this West Coast Computer Faire was a pivotal moment. For personal computing in general.

Bricklin: 3 April, 1980: "Mitch Kapor called. Went over the newsletter. TRS-80, no. Buy one if cheap. They're not going to send us one. We should go buy one. Ed Esber called and said, 'I need a dealer demo for the Commodore to take to Hanover Fair.'" So we were back and forth providing things for them.

Ceruzzi: You were furiously porting.

Bricklin: Yes, this is when Seth Steinberg is doing his thing. This is before the PC – before they came. "Met with Julian Lange of HBS." This was 4th of April, 1980. "Met with Julian, told him everything that's confidential. So he said, do real financials with appreciation projections. Cash flow. List assumptions and implications. 1981 to 1986 goals, strengths, weaknesses." This is all the stuff together with them, without them. This is standard how you do a merger. We were discussing merger. We kept doing that. "Julian recommends less speed but showing strong interest." This is teaching us negotiation techniques.

Ceruzzi: The merger, did you automatically assume this was going to be good for you to do a merger?

Bricklin: Well, obviously, we wanted to get this thing going and it was exciting.

Ceruzzi: You didn't think of yourselves as kind of going public on your own?

Frankston: The merger seemed to make more sense. Even if we went out on our own, we had the contract locking us together, so it seemed to make more sense to work together than fight.

Bricklin: "Let them make an offer; don't commit on price yet" . . . stuff like that. Basically we were learning how to do negotiation. We're really talking about this. "Questions for Mitch Kapor" on the 9th of April "Atari copy, have them call about this. Try other VisiCalc's to see if it

boots. Updates to our publication SATN 22. It's open whether it's DOS I or not. Bugs. Called Stansfield at Radio Shack; left a message. We had to check out memory allocation, because we were working on Radio Shack. This is stuff for Seth."

Ceruzzi: This is a merger with whom?

Bricklin: Oh, no, so – we're . . . in the midst of all this, we're still porting as you say. But the merger was going to be Personal Software and Software Arts.

Ceruzzi: What had Personal Software. . .

Bricklin: Oh, it showed VisiCalc to Stan Olsen at DEC, and Jerry Moore on the 10th of April, 1980. You got it.

Ceruzzi: That's my question right there. 1980 is a little late for DEC to be getting . . .

Bricklin: Here we talked to Julian again. "Get them to say what they think." This is all about negotiating a merger.

Ceruzzi: PSI – what products did they have at the time that made them rich?

Bricklin: MicroChess.

Ceruzzi: Oh, that was Peter Jennings, right?

Bricklin: Yes. He was their chairman or vice chairman.

Ceruzzi: Did they have anything else besides that?

Frankston: Well, the timing . . . once they started naming things Visi . . . oh, they did: Tiny Troll, which became VisiPlan/VisiPlot because of Mitch. Then they picked up . . . was VisiDex . . . (?) no, that was Dave Weiner – no, actually, no, they already knew Dave. We didn't produce it, we vouched for them.

Ceruzzi: Dave was with Living Videotext.

Frankston: Yes, right.

Bricklin: He came in and showed at the same time we showed at the West Coast Computer Faire, privately VisiCalc; he was demoing to a select group of people his first version of his Outliner

program. He got to show it to Ted Nelson, who didn't like it, which was devastating to poor Dave. Dave went on to bigger and better things, so it's OK.

"11th April, 1980 – met with Dan Fylstra and Hank Smith." He was from Venrock, one of their venture capital investors. Dan, I guess, had picked up venture investment. "Discussed the goals, etc. for the merger." So we were still doing this some more on the 13th. "Went to Wang with Dan and showed them VisiCalc. They want it and Dan's working on a proposal."

So constantly all these new machines were keeping us busy.

Ceruzzi: Wang was here in Boston.

Bricklin: Right, so Dan was here for that.

Frankston: We hired their librarian. Actually, she went to Wang for a week and then we got her back. She was Paulyn Heinmiller who later became the Lotus librarian and then the Microsoft librarian and archivist.

Ceruzzi: Wang ended up going with Multiplan – is that right?

Bricklin: That was later. I don't know what the negotiations were with that? Did we ever do a Wang version of VisiCalc?

Frankston: I don't think we did. But actually, Ray Ozzie's wife worked at Wang.

Bricklin: "Mitch talked to somebody at Apple about two methods of supporting the new version of the Apple operating system. Apple had the new operating system and it had to deal with copy protection.

Frankston: Source.

Bricklin: "Julian Lange set up a timetable; he set up an agenda." We're trying to do this merger thing. He said there were "five things to do: do valuation of the company and VisiCalc – high, low, medium forecasts."

Ceruzzi: So what was the outcome?

Bricklin: What happened was on 5 May, "I asked him where we fit in and he said he wanted to talk about Terry Opdendyk. And our role depended on what needs to be done." Dan was hiring this president who's now going to be in charge and how they're going to do their stuff.

Ceruzzi: President of . . . ?

Bricklin: Of Personal Software. So that was 5 May, 1980, when we're talking about that.

So now this is up to …. What happened was that eventually Terry came and there was no talk of merger. We're little nothings and stuff like that, and let's lower our royalty rate. We were devastated. At that point it was clear that we had to change our business model because we couldn't just improve VisiCalc and bet our whole company on VisiCalc because the business model was now different. We had to change our business model. Then we hired Julian Lange in as president, eventually president and COO.

We were going to have to do our own products and probably have to get our own publishing because the relationship at that point soured. We can go over all the incidents . . .

Ceruzzi: Not necessary.

Bricklin: But there were others. Mitch at that point had the same problems. He was an author and he left because of the same things and he was telling us. So there are all sorts of notes I have about how the relationship deteriorated.

I no longer was interacting with Dan Fylstra any more. I was interacting with Terry or with the product manager. Eventually our product managers talked to their product managers. So that's what ended up happening.

Then we started talking about a buy-out because it got so bad. Buy out our royalties. They were looking to buy out and so we started negotiating with them. Well, the principals didn't get along – Terry couldn't get along with them and they couldn't get along with Julian. So we said: Okay, Bob, me and Julian would not talk with their top people, but we had our next level down do the talking.

Frankston: Tracy, for example, was very competent.

Bricklin: Tracy Licklider and Lisa Underkoffler, who was product manager for VisiCalc. The two of them would meet with whoever the other side provided. Originally they had a marketing guy, Rich Melman. So we started having negotiations over that. Whenever we thought we had an agreement, for some reason the person on the other side would disappear. We'd remember agreements that we agreed to, but Dan says he never heard of that. Clearly, there were some communications problems on the other side.

In addition, since we were starting, we cared about having our name being on it. So we asserted ourselves from the PR viewpoint. We also started a publication called SATN – from Software Arts,

the creators of VisiCalc. All this legal stuff about how to do it. At one point Melman, I think, threatened a lawsuit if we said VisiCalc too much.

DIVORCING FROM PERSONAL SOFTWARE

They were very clear that they wanted to have the name VisiCalc only associated with their company and not associated with us. At one point Terry called us "little boys" for saying that. So you can see where animosity entered the thing because they were trying to make the Visi prefix exclusively theirs. So eventually there was a lawsuit.

Ceruzzi: Was Mitch involved in any of this?

Bricklin: No, Mitch was gone. He had something between him and them about what he was allowed to do and he ended up starting to do 1-2-3.

Ceruzzi: Because from what I know of Mitch, he's a very good person – a people person, and he might have been able to smooth this over.

Bricklin: No, no, no. We couldn't go through Mitch. His feeling for Terry? "Don't have any friendly chats with Terry – Watch out for friendly chats with Terry."

Ceruzzi: No. But he is a people person.

Bricklin: Right. He is a people person. I'm just saying what people said back then – what we felt back then. What we feel now is another thing.

So we were trying to create a company because being associated with this great product had great good will – worth more than any royalties. They were trying to produce this other product. Now, the question is: why did they do the lawsuit?

Ceruzzi: The other product being . . . ?

Bricklin: VisiOn, a Windows-like operating environment with a calc product, a word processing product and other things. We didn't care about VisiOn. We let them try that and we would like to write for it, we even talked about it.

Ceruzzi: You didn't find the idea intriguing?

Bricklin: It was intriguing since obviously we'd seen the Xerox Star and we knew about it and the Alto. Remember the original VisiCalc was supposed to be used with a mouse.

Ceruzzi: Really?

Bricklin: Yes, it was supposed to be. So we knew this stuff! We had a laser printer in house. We had our own laser printer. You couldn't get laser printers. This is not the days of getting a laser printer from HP! We had laser printers in the early, early 1980's. We had an Ethernet, a 3 Meg Ethernet.

There was a point, I think this was at SoftCon the 27th or 28th, a Sunday. It was late at night. I ran into Dan Fylstra in an elevator. "We discussed additional VisiCalc features and I asked for guidelines to measure them by. We danced around a long-term solution and he said we have to go our own way as a deal. I said that this was a possibility" which was what we were trying to do! We were trying to negotiate that type of thing.

"But I felt that we could work out a deal. Also it was possible to continue cooperating like Microsoft and IBM and VisiCorp and DRI at the time, which were still cooperating. He said he'd like to have a 3 month deadline. I said I'd like to do it. We had tried doing it many times before in X months and always something would happen."

Normally, the person we negotiated with disappeared! He was no longer working for the company. We negotiated with 3 different people. Melman and Ron Fisher and Tom Towers or something. "I told him to give Tom Towers the power to mold a formal agreement which he hadn't had so far. Dan said he hadn't seen any proposals from them. I expressed surprise. I thought we almost had a deal several times." I told him that back then. "We talked about the bad non-months (?) – the whole thing I just talked about – ever since. the merger discussions where Terry put us down; it tainted our relationship ever since. But I told him we could mold a deal that would be better for all but you have to try."

But the thing is – "our relationship is valued" . . . and back and forth with this stuff. He told us how worthless the VisiCalc name was becoming.

Ceruzzi: Why?

Bricklin: I guess 'cause we didn't improve it, I don't know. I told him that they should give it to us or get rid of it. He said that's something they wouldn't do because they valued the name VisiCalc.

Ceruzzi: He was saying it's worthless, but at the same time he wouldn't give it back to you.

Bricklin: Yes. "He mentioned something about 'ending up in court sooner' if we had done something. I said we hadn't done anything and then he covered himself." So clearly, he was

thinking lawsuit then. We weren't thinking lawsuit! Why would we bring a lawsuit, especially because around that time we had starting negotiating for selling our company. We didn't want a lawsuit in the middle of negotiations. By September we were about to bring it to a close.

Ceruzzi: Selling it to whom?

Bricklin: We were about to sell our company to the CompuServe division of H&R Block – and for cash. We were about to close if we could pass muster. The Boards had okayed it. We had okayed due diligence from Goldman Sachs and we were about to close the deal and a lawsuit shows up.

Now, according to Dan, he says we threatened to sue him, though I have a record of at least twice when he'd threatened to sue us. They said we were trying to stop shipment of VisiOn. I said he must have heard that from somebody.

Ceruzzi: Oh, VisiOn had already been developed?

Bricklin: Yes, they were developing it. In fact, in this discussion, the reason I've mentioned it is – we talked about it. "That we will be the first VisiOn ISV. I was pleased about it. He felt that VisiOn was ten times better than anything else. He said he needed a VAX. I told him it's possible for us to do that and we're considering buying a VAX anyway." So we were discussing becoming developers for VisiOn.

Ceruzzi: Did he hire other programmers?

Bricklin: Oh, yes, he had a huge staff. He hired them on the west coast. They put together a whole development group instead of us. So instead of merging, they did this. So at this point he was getting involved in management again. If you look at what he wrote, it doesn't seem to fit with his story of "out of the blue we were going to sue him and he had to protect himself."

The other thing was – the lawsuit came down to . . . we got the trademark back not because whether Dan or Bob made it up – it didn't matter. What mattered to the court was the clause in the contract that said under certain circumstances of termination we got it back. And we used the best efforts clause because they weren't marketing VisiCalc. They weren't spending anything, not even a token amount compared to the competition.

There was a reason for that, because by the spring of 1983, 1-2-3 was eating everybody's lunch. Multiplan was being eaten. We had done a VisiCalc Advanced Version which we shipped on the Apple II, Apple III and later the Apple IIE. We were converting it to run on other machines in our implementation language.

Almost our entire company was based upon making this implementation language work on all the machines of the day: the DEC 350 which they asked us to produce VisiCalc for. The IBM PC which they asked us to produce VisiCalc for. Who was creating the run time for that? Who was doing an implementation language for the IBM PC? Ray Ozzie. We had really good people. We had these great people developing VisiCalc and we were writing a VisiCalc and VisiCalc Advanced Version with these people.

We had two developers working on our TK!Solver product. A lot of the rest of the company was developing the marketing for TK!Solver.

Ceruzzi: What was TK!Solver?

NEW PRODUCTS

Bricklin: It was a back solving tool which let you put equations in and it would do back solving so you could say – x is . . . and it automatically did unit conversions. It's still being sold. But we realized we had to go our own way, so we started doing that and we shipped it around the same time 1-2-3 shipped in early 1983.

Well, first of all, we DID do CP/M versions. The reason I'm doing this [all this reaction to Dan Fylstra's essay] is that – this is not a normal 'about the history'– I realize that if something is in the record, it gets repeated, like the thing that said we got venture capital . . . no, we borrowed . . . and things get wrong about who did what. And if you don't put it right . . . and I think it makes a difference to understand the dynamics that led to 1-2-3 being able to get there.

Mitch knew what was going on between the two companies, so he knew what we were up to. He was a beta tester of VisiCalc Advanced Version which had a lot of the features, but he did it his way, his features, but he knew a lot of the features. But he also was a beta tester for Multiplan and for other things.

Ceruzzi: Did he ever have any discussions with you about his emphasis on the IBM PC?

Bricklin: No, we didn't know what he was doing. Maybe he told us he was working . . . but I didn't know what product he was doing until a month or two before he shipped when he showed it to me. I knew about his company; I knew the name of the product and I knew some of the stuff. He was relatively open. Not as open as Vern Rayburn would be about stuff, but he was very open about stuff.

So he knew about that, but he knew about other products. Now – this IL implementation language: we were converting it to a higher level language to run on these faster machines. It turned out we

were slow so we were working hard to get this VisiCalc Advanced Version – to make it work. The original VisiCalc was lightning fast. Mitch told me when he first showed me 1-2-3 – he first showed Context MBA, which was slow, written in UCSD Pascal, and then he showed 1-2-3 which was lightning fast. He said: Dan, I remember how you told us how Frankston made it go as fast as the repeat key. And we worked hard to make it that fast which is why it's in assembly language. VisiCalc was that fast.

Of course, it [VisiCalc] didn't use binary arithmetic. It used decimal, but it was still pretty fast. There was a product that was 20 times slower or 10 times slower called Context MBA. That's what they sold against theoretically. That was the straw horse.

Ceruzzi: Context MBA was an IBM PC product?

Bricklin: It was a product for the IBM PC written in UCSD Pascal from a company called Context; the MBA product had similar features to 1-2-3. So they took it on because they thought the integrated product was the way to go. The people only cared about the spreadsheet to start.

Ceruzzi: The 2 and the 3 were forgotten.

Bricklin: No, they were extremely important long-term to the industry because of the graphing . . .

Ceruzzi: 2 was graphing?

Bricklin: I believe that the spreadsheet . . . first, some spreadsheet is better than none. Going to 1-2-3 – the advantage of 1-2-3 over VisiCalc was better presentation. There were a whole lot of other things, but what people cared about first was: it had variable column widths. Ask Vern Rayburn and others who demonstrated it to people when it first came out. Variable column widths. Commas in the numbers – so they could actually use the output – and long labels. So they could actually use the output directly in a report.

The original VisiCalc – you really couldn't use the output directly in a report. When we went from 1-2-3 to Excel, we had laser printers and you could use the output directly in a report. 1-2-3 also had graphing, which you could use in a report and you couldn't do that by hand so you went from no graphing to some graphing with 1-2-3. Excel gave you a better looking graphing.

It also had a simple database, and many people used VisiCalc and other spreadsheets as a database.

Ceruzzi: We had that discussion last night, that Excel is actually the most popular database in the world.

Bricklin: Right, and 1-2-3 was before it, even without using the database commands. So the 1-2-3 was being used; it had word processing sort of, because it could do wrapping, but people didn't use it for that.

CP/M, I remember we did CP/M versions, but I remember the fact that it didn't have copy protection was a real problem for the people at Personal Software. It may have been one of the reasons why they didn't do the Osborne version. But I think they may have been also because we didn't have the time – we weren't able to get it done in time. I have some of that in my notes in terms of why we didn't – the decision to do the Osborne version.

Ceruzzi: When Osborne was sold, what was bundled with it?

Bricklin: SuperCalc, WordStar and dBase. And SuperCalc was first demo'd on the Osborne. I have photographs to show you that, when it was first in development. As Frankston says, unfortunately I asked them about some features which we had that they didn't have and they said – Oh, why should you do that? And unfortunately I told them. I thought they had looked at VisiCalc before cloning it!

Mitch knew VisiCalc through and through – knew all the good things, and copied the ideas that were good and threw out the ideas that were bad.

Ceruzzi: What was an idea that was bad that he threw out?

Bricklin: Not being willing to do certain things about long labels and certain other things, and using C for copy instead of R for replicate. Oh, and the way of doing replication – indicating between relative – and absolute – not in the command but in the formula itself which Multiplan had done before. I don't know if he copied it or just came up with it, but he did the right thing.

Ceruzzi: But the story goes that the real crucial difference was that he went with the IBM PC and you were still with the Apple III.

Bricklin: No, no. As Dan Fylstra said, we sold more copies on the IBM PC of VisiCalc than anything. We supported, unlike what he said here, we supported more than 64K; we supported 512K because they only gave us a 512K worth of memory – we never got 640K. So there was a bug above 512K until we actually got it to fix the bug. It looked weird because the numbers wrapped because we only had two digits, but it actually initially supported more than 64K by the time we shipped it or within the next release.

It was shipping an incredible number of copies. In fact, one of the reasons VisiCalc was still making money is that IBM continued to buy their copies. The reason that VisiCalc was still selling

is that first of all it was on the Apple III. We won awards as most popular product on the Apple II, on the IBM PC in 1982.

VISICALC ADVANCED VERSION

Ceruzzi: And this IBM version was written in this higher level language?

Bricklin: No. It was the original VisiCalc. It was VAV – VisiCalc Advanced Version, VisiCalc II, that never made it into the IBM PC.

Ceruzzi: So the original version was ported using this Assembly technique that you developed?

Bricklin: Yes. We basically took the CP/M version, which we had been shipping. The day the IBM PC was announced was the same day that the HP 125 Apogee (the one with the butterfly) was announced. We were on that machine with the CP/M version and that version was used for the conversion to the IBM PC.

IBM would not ship their machine without knowing VisiCalc could ship. It shipped within a week of IBM's announcement; it was in their ads. I can show you pictures of Charlie Chaplin and all that. VisiCalc shipped hundreds of thousands of copies on the IBM PC, many of them sold by IBM, who bought them from Personal Software/VisiCorp to sell themselves.

So VisiCalc had a lot of hysteresis because it was on other machines. So first of all, not being on CP/M, it was their decision and part of it had to do with copy protection, I believe, not just with speed. There are a lot of reasons why, and CP/M didn't matter. It was for small business and wasn't being used as a personal computer. So it wasn't as important to be on CP/M.

Ceruzzi: The Osborne had CP/M.

Bricklin: Right, but that had a memory mapped screen, so we could have done okay like SuperCalc did.

Ceruzzi: So the KayPro also used CP/M.

Bricklin: Right, and we could have been on those machines, but for other reasons it was decided against it. But at the time I told Dan about 1-2-3, although I wasn't allowed to tell him about 1-2-3. All I could say to him was, "Dan, Mitch is doing something; I don't know what it is; find out what it is." This was at the Alex. Brown Conference in 1982.

Ceruzzi: Alex. Brown?

Bricklin: Yes, that was the big conference for investors and I got to speak. Even though we weren't looking for money, I was the last speaker one year and I gave a history speech. One reason I have all this history is – starting then, in 1981 I think or 1980, I gave a talk at Alex Brown using slides. I said I'd tell them what it was like to be in a small business. Here's the history of my company; I have all these slides. Some of them are on my web site. So I had gone through this history with that and every year I came back and talked about personal computers and why personal computers are important, because they were all into minis and mainframes.

But eventually we became important and the venture capitalists cared about that. I saw Dan and said, "Dan, you have to see what Mitch is up to." And Dan said, "Mitch is a nobody."

So eventually Lotus sent the invitations out. The day the invitation was sent out, I had seen 1-2-3; Mitch showed it to me. I said, "They'll sell $60 million the first year." I was off by $7 million apparently. Mitch thought they'd sell $3 million. The business plan said I was close. I was blown away. Thank God we had another product and we had at least some income if TK!Solver didn't do so well, but we didn't know that.

The invitations came out, and now I called Dan to tell him I saw the product and described it to him. His reaction as I recall was – I think we can take it on with Quasar. He didn't care about VisiCalc. Quasar was the internal name for VisiOn.

Ceruzzi: If you think about VisiOn as a Windows before Windows, it could have happened.

Bricklin: That's right and I felt fine with it, but it was just that he had given up on VisiCalc. So basically a lot of the stuff about the destruction of VisiCalc – he says the lawsuit didn't hurt because sales still continued.

Ceruzzi: Lawsuits do hurt.

Bricklin: They do hurt. He said it didn't affect the sales because of the hysteresis; they kept it quiet. He said the thing that did it was the anti-marketing campaign of Software Arts. Because in the middle of the lawsuit we finally said, "Okay, we're terminating the contract." We got new lawyers, very aggressive lawyers who said, "Let's take back the contract and we'll start selling it ourselves." We would sell it to the dealer channel at normal amounts, but we would sell it now direct at $99 because how else could you take on 1-2-3 except with a cheaper version?

We took VAV, bundled it with the real VisiCalc and put it in a box with a book about how to learn VisiCalc. We put that together and we came out with that. Dan says that's why VisiCalc died. This was at the time when 1-2-3 had been on the Softsel hot list for over a year and had sold at that point

probably around 80 to 100 million dollars worth, compared to VisiCalc which had sold a fraction of that. A lot of that was IBM selling through IBM and others. So to say that it was the anti-marketing campaign that killed it as Dan Fylstra does in his essay, given that the VisiCalc name still was valuable, it's kind of wrong.

He says that this injunction period (which lasted four days not one day) because Bob had a sworn affidavit that he coined the term VisiCalc. Maybe there was an interrogatory and Bob had to say who did it so he said he did – I don't know. But he said that his feeling here was that the reason it was granted "with a burden of proof and waters muddied by language of publishing and Frankston's sworn agreement that he'd coined the name" – that had nothing to do with it. It was the not using best efforts in marketing. He forgot that they hadn't been marketing it.

Ceruzzi: Was there ever any bad feelings with Mitch about his coming out with 1-2-3?

Bricklin: We had mixed feelings with Mitch because we had not put graphing in our product because Mitch was doing a graphing product so we didn't want to compete against him. He wasn't going to do spreadsheets I guess. But it was very mixed, because he did the right thing in hindsight and it was good for the world in the end, and because he is the one who actually purchased our company and saved us from bankruptcy. That was really good. When others wouldn't – we went to everybody and it was a real pain. So Mitch was willing to, and he did it in five days from when I first talked to him till we had checks in hand to bail us out at the bank.

Mitch treated us well in terms of that. So competing with us – it hurts when somebody competes against you who's a friend who you work with. It makes you feel good, and they took everything that you told them that happened in hindsight to probably be good and used it.

Ceruzzi: He didn't steal anything from you.

Bricklin: He didn't steal anything. He was a beta tester, which bothered us. Here we had a beta tester who was developing a competing product. On the other hand, in hindsight, compared to other things, that wasn't that bad.

So it was hard, but we kept up cordial relationships, and the fact that we did, helped. What I learned was, the fact that I talked to Mitch saved our company. There's a story about how I ran into Mitch at an airport because in the midst of all this, I didn't go to Mitch. That's the one thing Fylstra says – If only he had called us. He did call me – a week before the lawsuit he called up and said – Dan, we really want to have a deal and I said – Dan, we really want to have a deal, too. He called me on the phone about a week before and I thought – Why the hell is he calling me out of the blue?

I called Hank Smith at their Board and said, "Can we find a way to settle this?" We had money which we could have used to pay to get out of this thing. I was rebuffed by Hank Smith. So he

went to the Board; they said no, it's being handled by them, not you. We don't know if somebody there knew the deal was coming out and they were going to scuttle it, because that meant we had money behind us and maybe the money behind us would be a problem for them.

Ceruzzi: The deal being . . .?

Bricklin: Us selling our company. There's the possibility that they got wind of the fact that we were doing it. Dan may not have known about it; but I don't know. The fact that we were selling the company for $50 million cash and that the lawsuit against us asked for $50 million cash surprised us. Also the date and a lot of circumstantial stuff surprised us, and the fact that when we approached their Board for settlement, it kept on being rebuffed.

Eventually we had to lay off half or 2/3 of our company; they had a layoff, too, and they fired or laid off their president, Terry Opdendyk. At that point they brought back Melman who had negotiated the original deal with us that we were going to settle with, and he talked to Julian and they came up with a deal within a month or so and we separated the company and we ended up with the name VisiCalc and they ended up with Visi and whatever else they wanted.

So the way the story goes – this [Dan Fylstra's] story does not fit with: (a) as I remember it; and (b) as I have stuff that goes with it. But I think we have to understand that 1-2-3 won on the merits of the product, and I believe it probably may have even been able to win against VAV directly, even if we had done everything VisiCorp had wanted us to do.

Ceruzzi: Why?

Bricklin: Lotus 1-2-3 was a good product. First of all, there wasn't a thing about the two machines. If we had only done the IBM PC first. They told us to do Apple first. We had to make a decision. And the decision was to do Apple first.

Ceruzzi: They being?

Bricklin: Personal Software/VisiCorp. It was their decision; it was up to them.

Ceruzzi: Not Jobs.

Bricklin: No. Not Jobs. There was a point I believe where Dan Fylstra told me a story that it got so testy between Fylstra and Jobs about something that Arthur Rock invited them both to dinner to try to settle them down.

Later on, Jobs invited me and Fylstra to a meeting in the middle of something to speak to each other. It may have been that day but Dan didn't show up at it. But I told Jobs what he should do and he

told me to forget it and whatever. I still think he should have done it, but it isn't Jobs' way of doing things. There would be no Windows today if he had done that – if he had done what I suggested.

The thing about 1-2-3 is that it was tuned to it [the PC]. Context MBA, Multiplan and VisiCalc all went with a strategy for all machines. Mitch was trying, but he was a small company and he had to put his resources and he put all of his wood behind one arrow. And he had a great programmer. Jonathan Sachs did a great job. For years people tried to knock that code base off on the IBM PC with 640K, but nobody could knock it off.

Ceruzzi: What was that – Paperback Software?

Bricklin: Paperback . . . oh, eventually with clones, but even that couldn't knock it off because why buy a clone?

Ceruzzi: It was so cheap – Paperback.

Bricklin: Lotus wasted their time. Lotus is a separate thing, but the Lotus lawsuit – it's Paperback. I was an expert witness for Paperback and I was in court so I got to hear what was happening.

Lotus in court said – "Your Honor! These clones are horrible! That's not the way to compete with us, Your Honor. The way to compete with us is like – Look at Excel. This is the way to compete with us. It's a different product. It's not the same. Look – it sort of runs our macros the same but not identical. It sort of has a conversion mode – ha ha ha it's not exactly right – but Your Honor, this is the way to compete with us."

Ceruzzi: So they wasted their time.

Bricklin: It turned out they were right – of course this is the way to compete with us! The clones never made that much money and it really didn't dent them that much. And when this other product came out, when Excel came out, that knocked them out. Excel was a better product and Excel had better visual output. It took advantage of the laser printer.

Ceruzzi: Which was because Excel was developed first for the Macintosh.

Bricklin: Right. And also Excel was first developed for the Macintosh and established itself as a brand, an established great product already. So now we know Excel was a good product.

Ceruzzi: And the Excel developers got the Macintosh religion.

Bricklin: Right, and then they moved it, while the Lotus people didn't want to help Microsoft with Windows at all. I was with Mitch when he first saw Excel. Similar to me, apparently he had some deal with Gates and knew about Excel and when he saw it, his jaw dropped. Mine did. We were at Comdex or something and we saw it in the booth. He said – "That's not what it was supposed to be!" Basically the same thing that Fylstra did when he saw what Mitch had done. What comes around goes around. I feel bad for __?__, because Lotus . . .

Ceruzzi: But Mitch also had a lot flashier marketing than VisiCorp?

Bricklin: Well, he spent a million dollars. VisiCorp spent nothing on marketing there. Yes, he did much better marketing. One of the successes was they did an exclusive with Softsel. So they jumped up the channel. They got all the best dealers and had them as part of working with it. You can talk to Sy Merrin about what they did. Talk to him and others how they did it. They executed the dealer strategy. I think they probably did the corporate selling better. Remember, they had Ben Rosen behind them and while Personal Software, as you can see, was into the dealer channel and doing that as opposed to the corporations. I guess he keeps talking about dealers and selling to individuals as opposed … this is beyond me, I'm thinking now in hindsight.

Ceruzzi: So those were things that you didn't do?

Bricklin: We weren't involved in any of that stuff. Our marketing at that point all had to do with TK!Solver. We didn't have a marketing arm. Our marketing arm for VisiCalc was VisiCorp/ Personal Software.

So just to look at this, his quick answers were: they paid us a lot of money but we spent a lot of money to be on all these machines and we had to port to every machine whether or not they wanted to. We knew this was going to end and we had to separate. My notes say we knew we were going to be out. Our whole business plan was based on that we were going to have to cap VisiCalc. At one point we said – We'll take $10 million cash and we'll give you VisiCalc, the source code and the name and go away. I remember talking to Melman and it didn't happen. He disappeared and a new person took his place. The next year we got $11 million in royalties from them.

Ceruzzi: Did you ever think about just starting from a garage or starting again from zero?

Bricklin: With what?

Ceruzzi: With something!

Bricklin: Oh, afterwards I started Software Garden and was very successful. But that was after. I consulted to Lotus for 4 months to help them with the merger.

It *[Dan Fylstra's essay]* says that "several factors enabled 1-2-3 to quickly outsell VisiCalc. Chief among them: brilliant design, brilliant implementation." And it says it "was able to displace VisiCalc with PC dealer channel with incredible speed, thanks to the marketing push of Lotus combined with the big marketing pull of VisiCalc out of the channels."

I don't understand how there's a marketing "pull" by us a year and a quarter later, selling it which was also available to dealers if they wanted it at the low price.

Ceruzzi: I'm not sure what you mean.

GETTING THE HISTORY RIGHT

Bricklin: But it is . . . but . . . I'm only saying this for those who are seeing this on the video, I'm reacting to a document that you may not have, but I'm speaking to anybody in the future when I'm probably dead and you're seeing this, I hope that people care about that. And you want to know what really happened, and that since we have a document saying one side, what you can get out of it, and it's very valuable I believe to have it. I want to make sure that that is on the record.

This is really weird to speak to a camera like this because on the day the IBM PC was announced, we had a videotape being made of our staff meeting, because we videotaped all staff meetings. And at that meeting I read all the information about the IBM PC to our staff because I wanted to tell them: this is what you're allowed to say; this is what not.

And I say into the tape: "Listen everybody, I'm doing this for people who aren't here today so people can see it later." I meant employees who didn't show up. That tape ended up . . . it was the only tape that the guys doing Triumph of the Nerds found.[3] And that ended up on TV years later – showing the IBM PC, and there I was saying "for you to see." Millions of people years later on PBS saw me talking and Julian Lange shows up on this TV program because I said that . . . what happened was we were doing all this research at Software Arts – lots of research, new products. And we had a UPI feed. So I said until it's public, you can't say anything. And somebody yelled from the back – "It's on the wire," because they had read the UPI thing that it was on the wire. So I said – "Oh, it's on the wire? You can say it, it's public, you can say it now."

And then Julian looked at his watch. And the camera had that. And the voice-over is – That's the moment when the IBM PC was announced. That timing – we never know what's going to happen, so for any of you who have seen videotapes of birthday parties . . . I've done this where I wrote for people. Little kids say – Why are you doing this? I say – When you're older, you want to talk to

[3] PBS documentary on history of personal computing (1996).

your older self? And people are starting to do that now. So I apologize to those who hear it who don't also have . . . this sounds like it's a diatribe against them . . . .

Ceruzzi: Oh, no, that's alright.

Bricklin: . . . . but it's trying to fill in – hopefully for historians. I'm trying to fill in the other pieces of what went on. I have notebooks with stuff if you care.

Ceruzzi: The notebooks to me are extremely valuable.

Bricklin: I found it cool to see . . . I'm reading it and going – Wow, look at the interactions between all these manufacturers. Some of them didn't have a clue and some did and how they're jockeying for position and what people were saying. It's really, really cool. And that was going on at the same time we were trying to build businesses.

Ceruzzi: And you're trying to write code!

Bricklin: And writing code and developing!

Ceruzzi: And I'm fascinated by this notion of the high level language versus assembly language and how that plays out in various systems.

Bricklin: This is real interesting because – Why did Gates knock Java early on besides the takeover of the world stuff?

Ceruzzi: Because it's a high level language?

Bricklin: No, well he uses high level language, but it's a threaded byte code language. That's what his system was – he wrote lots of stuff. And he was blown out of the water by a better system for performance like 1-2-3. So there is some stuff there. That's why a lot of us questioned certain things about Java – why it was the way it was. Because with the machines of the time, it would make a difference. And why we still use C and C++ to do certain code. But you have to get the right tool at the right time which is why it's so important what Frankston did.

The thing was – as Bob shows – he was all over the place in abilities and what he did, but he was able to rise to the task and Seth Steinberg with the conversion. We also had Steve Lawrence who actually helped finish Bob's code in the first version. But the fact to actually craft a complete product to fit with the right feature set in the machine of the time is important. And that's important going forward and we all need that because we look to what Longhorn's [code name for Windows Vista] gonna be and stuff like that.

Ceruzzi: I guess maybe Java . . . I mean, nowadays you have this essentially free memory and tremendous processor speed, so you can waste a lot.

Bricklin: You can waste but you can spend it for other things, too. Then you can't spend it on other things. So which things did you spend it on? It's where do you put your resources is what this country is going through.

VisiOn was designed to run on the machines of the day. They would always show it on the Wang, because the Wang ran on an 8086 – twice as fast as the PC. And it was underpowered and they killed their cash cow, and suing us guaranteed it. Once you're in a lawsuit, you're guaranteed it's going to be a problem.

So they had no cash cow and of course it took awhile. The machines were underpowered for VisiOn. It was very text-oriented and it was all in bit-mapped text. And rendering text is very expensive. So a lot of things were very slow in terms of that. I got to play with it very little.

Ceruzzi: But you had visited . . .

Bricklin: I knew about that type of system. Also, we had seen the Macintosh. We developed one of the first products in the Macintosh – TK!Solver for the Macintosh. It came out because we were doing it in our implementation language, we got a version for the Macintosh up like this [snapping his fingers] of our product.

Mike Boich who was an evangelist for the Macintosh, I have all sorts of notes about Joanna Hoffman talking to us about the Macintosh – Mike Boich coming over. Mike wanted to do a summer job with us because he was between years at business school. He was a student of Julian Lange. We were Mac developers eventually. They wanted us to commit, but we weren't able to commit to be a developer like Mitch and Gibbons and Gates did and only Gates really followed through in a way that mattered.

Ceruzzi: With Excel?

Bricklin: No, with Multiplan, Word and Basic. Initially, when it first came out, and then kept going with Excel. Jazz turned out not to be the right thing. Lotus was more worried about packaging and marketing and stuff like that. And Microsoft went after what's the best Macintosh product – a spreadsheet, just a spreadsheet, not integrated.

NEW DIRECTIONS

Ceruzzi: The Mac already was integrated with its look and feel interface.

Bricklin: We had another problem as a company in that to diversify we made a mistake. To diversify we did our own building; we renovated our own building and put a lot of effort into making it programmer heaven.

Ceruzzi: Where was that?

Bricklin: This was on Mica Lane in Wellesley, Massachusetts. And actually the only money I took out of VisiCalc at the end other than salary was the money I made on selling the building. We didn't tend to our knitting but it was a normal thing to do. You buy your own building and put your company in it. We were a privately held company with no outside investors, so that was the right way to be able to use our money.

Ceruzzi: Well, there's nothing necessarily wrong with that.

Bricklin: Nothing wrong, except we didn't stick to our knitting; we should have put even more effort into VisiCalc. But our business plan, we couldn't put it into it because the relationship with the other company was so tainted – they kept saying how they needed to lower the royalties without giving us any reason why. It was just like . . . Well, your product is just like all other products.

Ceruzzi: It wasn't!

Bricklin: Yet they would say, "We will go to lawsuits if you misappropriate the trademark because we need the name 'Visi' on all our products." And I asked Dan at one point – and we talked about this last night because he remembers it – in the middle of one of the attempts to negotiate for a settlement in the middle of the lawsuit, I said, "How much is the name 'Visi' worth? Is it worth $100 million?" "Oh yes," he said, "It's worth $100 million." I said, "Is it worth $1 billion?" "Maybe," he said.

He still remembers that – that he viewed the name 'Visi' and that stuff. Now, the name VisiCalc, we only cared to be remembered as the developers of VisiCalc so we could be remembered as that. I think Visi turned out as Satellite Software which ended up with Perfect this and Perfect that and Multi this and Multi that. Those names didn't help as much. Why did Excel do better than Multiplan? Names don't work that well compared to product. Google – we worry about the name ... will the world accept . . . or Yahoo. Or Lotus. How silly. Lotus. Apple. And yet it's the product. The product really, really matters.

SEEING THE ALTO

Ceruzzi: I know this is sort of off the topic, but you mentioned seeing the Alto.

Bricklin: I saw it at MIT.

Ceruzzi: Did the scales fall from your eyes when you saw the Alto?

Bricklin: For me, I had been in computerized typesetting where they had these display terminals. We had seen WYSIWYG stuff. I had seen Englebart's stuff with the mouse. I worked with [Andy] van Dam one summer and I had seen Press and their type of stuff. So to me it was natural progression. I met Alan Kay. There's stuff here about when I went out and met Alan Kay. I had seen his stuff and I was inspired by Alan's stuff before I went to school. So that type of stuff – to me it's great. Normal progression in terms of stuff. Remember the mouse – VisiCalc was supposed to be done with the mouse.

Frankston: I was going to comment, again, out of context. I remember visiting the Spring Joint Computer Conference in Boston in 1966. I went to MIT, saw a 2250. 2250 was a remote graphics console for IBM. This was in the mid-1960s

Ceruzzi: Oh, yes, I do know that.

Frankston: And Andy van Dam was doing remote PDP 11's. We used the Space War with the wooden paddles and flying around. The ARD's terminals which were storage scope graphics terminals. The Imlacs with the PDP 8 built inside which had all factory debugging. It let you know you had a bug and you had to replace the board.

I remember [J.C.R.] Licklider playing with the Imlacs.

Ceruzzi: Is he related to . . .

Frankston: He's the father and he's the Licklider, the acoustic psychologist who got BB&N into the Internet. It's his son.

Bricklin: Tracy is a good guy. He's the son . . . .

Frankston: Right, he was here yesterday.

Bricklin: He was the Chief Operating Officer of the company.

Frankston: And that's what DARPA did.

Ceruzzi: But if you were never given DARPA money?

Frankston: No, this was totally unrelated.

Bricklin: I think we talked about trying to get an ARPANET connection at some point.

Frankston: Yes, we had email and everything.

Bricklin: Bob, let me give you an example of how cutting edge we were. We had a videodisk unit in there. We had an Aspen disk from the Media Lab.

Frankston: Yes, it's progressive and all that.

Bricklin: Bob encouraged this. Bob wanted us to have all this stuff. This Bob – Bob Frankston. After this all was over, I decided I wanted to work in my own little company and founded Software Garden. I didn't want to go to Lotus even though they offered me to go there. I would have gone there to work with Mitch; I stayed working with him, but I knew Mitch was . . .

Ceruzzi: Mitch left the company.

Bricklin: Mitch would not guarantee me he wouldn't leave. And [Jim] Manzi wasn't exactly my type. So we had problems with negotiations with Manzi. Mitch got between us as a buffer. Bob went there. Bob will tell you about how Lotus went.

Ceruzzi: Went to where?

Bricklin: To Lotus. He stayed there. Lotus.com was a machine

Frankston: …on my desk for two years. It was before the web. It was an email gateway.

Bricklin: And you were at that magic meeting with Gates.

Frankston: Yes. The thing I want to convey is how much of this was normal in the 1960s. I mentioned that story about Metcalfe sat next to me in class and about putting lineout on a coax, this was in Mike Hammer's seminar. So it's hard to remember without name-dropping. And I mentioned how Jerry Saltzer did this program called Runoff that eventually led to html. And the UNIX people came out of the Multics project. So these were just normal things. I was used to having these so-called mainframe vs. PC discussions.

But to segue into the Alto story. So one of the people – you weren't here, so I don't want to repeat too much, but at White Weld, when we worked on the SDS 940, it was the Berkeley project Genie machine, those people went to form Xerox PARC, but Lampson was a grad student consulting for us. So when I went to California in 1971, I wanted to say hello to him while I was there, and he showed me the graphics screen he was working on and this was pre-Alto.

Ceruzzi: This was at Berkeley?

Frankston: No, at Xerox PARC.

Ceruzzi: He had already moved to Xerox?

Frankston: Yes, this is after Berkeley Computer Systems had failed and everything. So, of course, this was my normal thing. So when Jobs showed me the Mac, I said – Nice price point; it's cheap, but I'd seen so many graphics terminals, it was not exactly new.

Bricklin: The stuff was a nice progression.

Frankston: But there was not enough memory on the Mac.

Bricklin: Bob being exposed to the stuff and seeing some things about it and also Dave Reed.

Frankston: And David Reed was the one who actually on the side did the Internet. The memory's less than the Apple, too.

Bricklin: The only thing wrong with the Mac – the thing we complained about was – no hard disk.

Frankston: And not enough memory.

Bricklin: And not enough memory. We begged them for more memory.

Frankston: The hard disk – I did complain, but the memory was like – Don't tell Steve but we could put 512K in this. But he snuck it in, despite . . .

Bricklin: And no arrow keys. Were there no arrow keys?

Frankston: No, don't you remember . . .

Bricklin: No, because we saw it early and we commented. We liked it.

Frankston: Those were Twiggy days when we saw it.

Bricklin: Bob and David – that's where I learned about stuff. So a lot of it – they would be exposed and little bits of it ended up in me and it's still that way. I depend on them to learn a lot of stuff.

Ceruzzi: But this other question, no one from ARPA came to you with money . . .

Frankston: No, of course not. In-Q-Tel didn't exist.

Ceruzzi: Right, but ARPA . . .

Frankston: No, but ARPA, Lick was already out of that. I mean, he was at Project Mac, but he worked with the cofounding of the Student Information Processing Board and stuff.

Ceruzzi: So was this a bias that they had? They didn't think that Apple II or . . .

Frankston: No – who's "they"?

Ceruzzi: ARPA.

Frankston: No. ARPA was just not an issue. Sen. Mansfield would want to know how many people we'd kill.

Bricklin: This was after.

Frankston: This was well after. The Cambridge project, and John Strayhorn was involved in that.

Ceruzzi: Did Xerox PARC try to hire you?

Frankston: No, none of that. No. Read the story about how they missed it. Xerox PARC and the Star.

Bricklin: And talk to Alan Kay about this.

Frankston: In any case, let me pull this together. When I was at Microsoft, I did not want to be in the research group. I realized research is just another job with objectives and rules. We actually had more freedom of meandering or careening in the product groups – and staying alive, despite the structure of the company – at least temporarily; you can ask Mike Maples.

But because I'm basically good at doing this… I try to have the answers before you have the questions. That's why what I'm more proud of than VisiCalc – though I'm actually pretty proud of VisiCalc after all these talks – is home networking – making that happen.

Ceruzzi: Yes, at Microsoft.

Frankston: And explicitly engineering that marketplace. Because VisiCalc was almost an accident. There was a lot there that we didn't expect. This time I said – You know, we've got to engineer a marketplace. This is why I'm now trying to fix the Internet but it's hopeless broken. And David Reed was a key. I was trying to get David to come to this and we should catch up with him, but he was the one who helped me understand the fundamentals of IP and "the End–to-End Argument" and we've been going back and forth on this.

Bricklin: I learned a lot of stuff.

Frankston: And this is why we go back and forth because David's now back working with Alan Kay on the current Kay project at HP which is the successor to DEC and he's also in the Media Lab. So there are all these streams coming across these influences, but the thing I want to throw out in a few minutes is how much existed back then.

I call it the Columbus effect. If you head in that direction and survive, you're a great discoverer. And in the 1960s, how much of what we call the PC stuff and stuff – we were simply learning this stuff in the 1960s, the QED Editor that ran on the 940. I can still use VI today.

Ceruzzi: I guess the thing that was interesting is that there were people at PC World who were rebelling against the time sharing, mainframe world.

Frankston: Oh, I was rebelling against it, but I was also . . .

Ceruzzi: You were rebelling against the notion of the IBM people . . . or the supervisor letting you know when you were allowed.

Frankston: Of course. And that's why we're rebelling against the Federal Speech Commission and why we have a Federal Speech Commission is an interesting issue.

Bricklin: That's the FCC . . . it's Bob's . . .

Frankston: Well, it's the same . . . it's the Federal Communication Commission. They're regulating communications, not technology.

Ceruzzi: I guess my version of the history is that the Apple II was sold as: you could use this as a doorstop if you want; we don't care. Nobody's going to . . .

Frankston: No, but more important to me was not the intended purpose. I'm used to reprogramming systems. What it meant is I'd go to the iron. We could rethink what it meant to go to the display. We could reinvent it and I've been doing that all along.

Ceruzzi: But you couldn't do that on an IBM machine.

Frankston: I could! I could. I was able to do it on an IBM machine. That was an interesting day, we re-did the operating system. We did things.

Ceruzzi: Why did you have that kind of access?

Frankston: Because we were . . . I was a systems people . . . . I could do what I wanted to do.

Bricklin: Like Bob said, if you're willing to stay up late at night.

Frankston: But also it was position. And including the times you needed a new instruction, then build the damn instruction.

Ceruzzi: And where was this?

Frankston: Two roles. The outside role I started at White Weld Computer Research Division, with the SDS 940. The thing of the SDS 940, it's a 930 but the 930 was an open up machine. I can see if I needed an instruction, I could do it, I usually don't bother doing it, but I had the option. We did the disk drivers ourselves, the interrupt level handlers. On the Multics they would get David Reed on the operating system, one project to do was take the first millisecond of booting and re-do it and do you don't use interrupts, you use processes. That was a kind of access. You read the microcode charts. Hey! Like when the 370 first came out, hey, this has pay__?___. They didn't tell us – let's use that!

Bricklin: It was a different attitude.

Frankston: It still is that attitude.

Ceruzzi: Did you meet Ken Olson?

Frankston: Not in person.

Bricklin: You met him when I met him. I got to talk to him at . . . were you there at Wizards and Wonders?

Frankston: Oh, that one. And we also met Eckert.

Ceruzzi: I'm talking about earlier on.

Frankston: No, I never worked at DEC. I was one of the few people here who never worked at DEC. I only programmed the PDP-8 with minor code. I don't know if I ever wrote a PDP8 program at all. I did a little PDP-10 programming when it was doing some CIT support under ITS.

Ceruzzi: Incompatible time sharing.

Frankston: Yes. But I was actually more in the IBM world with system programming and PL/I and the Multics world. There's Prime.

Bricklin: We went to the Prime because . . .

Frankston: Because of Multics – it was Multics-like, but not enough. Because Max didn't quite get it right.

Bricklin: It had segmentation which made programming easier for the type of stuff we did.

Frankston: Yes . . . . the Emacs, the QED, which I started writing an interactive editor and it turned into an Emacs.

Bricklin: And we couldn't get a DEC machine and ran UNIX ourselves in those days.

Ceruzzi: Why?

Frankston: Well, UNIX wasn't much of an operating system in those days.

Bricklin: First of all, it was a much more expensive machine in certain ways, but it was just not commercial.

Frankston: Well, we wanted PL/I, the language in Multics and it looked like it was going in that direction. It was a poor man's Multics. More than UNIX was.

Ceruzzi: That was the IBM . . .

Frankston: No. We didn't use the IBM PL/I. We used Bob Freiburghouse's PL/I. Later Stratus. But in my mind it was a Freiburghouse PL/I. Bob Freiburghouse – another interesting character.

Bricklin: He did PL/I for Multics. Then he moved it to five other machines.

<END>