# Oral History of Roger Sippl

Interviewed by:
Luanne Johnson

Recorded: December 13, 2005 and January 12, 2006
Mountain View, California

CHM Reference number: X3655.2007

# Table of Contents

# Roger Sippl

## Conducted by The Information Technology Corporate Histories Project

**Abstract:** In this interview, Roger Sippl describes his career prior to founding Relational Database Systems (later renamed to Informix) in 1980, including developing the design for the INFORMIX relational database management system while employed at Cromemco. He describes his conviction that UNIX was ultimately going to be a significant platform for commercial users and his goal that INFORMIX would be *the* relational database for Unix platform. As UNIX was not seriously considered by most commercial users in the early 1980s, Informix had to heavily promote the use of UNIX as well as promoting their own product and Sippl covers how the base of commercial end users developed during the 1980s. He covers how the company used contracts with OEMs to obtain large infusions of cash since they were not able to obtain venture capital funding in the early years. The merger with of Innovative Software of Lenexa, KS in 1988 is discussed including the resulting disruption to their business created by trying to merge the two companies which ultimately led to the decision to bring in Phil White as CEO.

### Background And The Decision To Major In Computer Science

**Luanne Johnson:** Let's start with your personal background. I know you graduated from Cal in computer science but tell me a little bit about where you came from and how you ended up at Cal.

**Roger Sippl:** Well, I grew up mostly in Southern California and I was attending the University of California as a pre-med and one of my best friends got into the optometry school at Berkeley. He got in after two years of undergraduate work which was pretty rare and so he was going to be able to do six years instead of eight to get his Doctor of Optometry. He was transferring to Berkeley and I kind of wanted to move further away from home and I thought the medical schools back east would recognize Berkeley more than _______________________, so I moved up there with him.

I was on campus only for a few days when I got sick. I kept going to the student health center. They kept sending me home with a little blue pill and finally I told the guy, "You've got to take an X-ray or blood test or something because I don't just have a cold." I seemed very run down. I had fevers and night sweats.

He took an X-ray and he saw a tumor in my chest the size of a softball and lots of others above and below. I remember he was looking at my X-ray and looked at the little metal plate at the bottom of them that had my name on it. He looked at that and looked at me and says, “Are you Roger Sippl?”

And I said, “Yeah. That’s me.”

The guy looked at me again to make sure I was still alive and then he said, “Did you fill out the supplemental health insurance form yet?”

And I’m like, “Oh, man, am I in trouble.”

So he sent me to an internist at Berkeley who had a biopsy done and determined that it was Hodgkin’s Disease. And it was a late stage. It was Stage 3B out of 4 stages.

**Johnson**_:_ But you had no indication up to this time?

**Sippl:** No. I mean there were these fevers and night sweats and I kept feeling run down. So tried to figure out where the world’s expert on treating Hodgkin’s Disease was because my numbers seemed to be in the 20% chance of surviving.

Henry Kaplan was at Stanford and he was the world’s expert on Hodgkin’s Disease, had done all the research for it. I got into his clinic and commuted from Berkeley to Stanford being treated for Hodgkin’s Disease for 13 months of surgery, radiation therapy, and chemotherapy. Really thinking I was going to die. I did not think I was going to survive that.

And then I started working in an immunology lab at Berkeley as part of the independent study program so I could stay in school and keep my insurance policy because my parents didn’t have any money. So I had to stay in school just to have the money to be treated.

In the process of doing that I taught a class on cancer for undergraduate study for 420 kids. It was a multidisciplinary class. The biology of cancer. The medicine of cancer. The economics of cancer. The psychology of death and dying, the effect on the family structure. The cost of smoking cigarettes to the American taxpayer. We had epidemiologists and economists.

**Johnson:** Did you put that whole curriculum together yourself?

**Sippl:** Yeah. The American Cancer Society had an outline but the goal was to get an undergraduate to put together a committee to invite all the guest speakers and develop the particular classes. And so we did that and I taught it although I had a professor who was the instructor of record but it was sort of my thing.

I taught two of the sections and I got the opportunity to start dating after all these years of throwing up all the time.  I was no longer in treatment and there were all these co-eds in my discussion section.

While I was doing all of that, I went on some pre-interviews at some medical schools and they told me that I wasn't going to be able to be admitted because at that time I only had a 50/50 chance of living when I got out of medical school.  They said they can't let anyone in over 30 years old because they have to deliver the most doctor service years per training dollar.  And I understood that.

They suggested that I switch majors and graduate, get a job, buy a car, date a girl, you know.  I kind of agreed with them although I spent a lot of effort trying to get into medical school and I think I would have made it had I not gotten cancer.

So I switched majors to computer science so I could get a job but that required that I stay a fifth year.

**Johnson:**  You decided on computer science because you knew there were going to be jobs out there if you got a degree in that?

**Sippl:**   Right.  My father had written books about computers going back to the early '60s.  He wrote the first computer dictionary.

**Johnson:**  Oh?

**Sippl:**   Which is in this museum somewhere because I saw it one day when I was here.  *The Computer Dictionary* by Charles Sippl.

I was one of seven kids that alphabetized various versions of that dictionary.  And he had started companies that were information bureaus about computer companies.  All of his kids worked at these companies so I knew about the industry and I knew computer programmers were in short supply and that you could turn a buck doing it.  So I figured that if I can't be a doctor maybe I can be a computer consultant.

## Employment At Bechtel And The Decision To Seek A Job In Silicon Valley

I was going to go get a Master's degree in medical information science at UC San Francisco to use some of my bio-medical education.  But during my fifth year, in order to put myself through school because all my scholarships had run out after four years, I was working as a contract programmer in San Francisco at Bechtel Engineering.

But I discovered that there was no need to go get a Master's in medical information science because just basic information science was sufficiently screwed up and surprisingly primitive and inelegant and uncoordinated. So I could easily make a living starting right then by helping people solve their business data processing problems.

And, furthermore, I realized that the applications I had been working on at Bechtel, which were keeping track of parts of nuclear power plants, which did not use a database management system – in fact, the process of database management was still pretty new…

**Johnson:** When about was this?

**Sippl:** '76? '75 – '76.

Minicomputers were fairly new and Bechtel was using minicomputers. There weren't that many database management systems that ran on minicomputers. And they weren't very full-featured. They weren't that usable.

So it dawned on me that there an opportunity. I went from one consulting project to another realizing that most of the code I was writing contained pretty much the same thing: storing and retrieving records on disk drives. And searching for records using various key fields. Trying to relate different tables together which at the time were not tables in databases, they were just files. Trying to relate these customers to these orders. How to relate these parts to these engineering submittals to these rejections to these reapplications. And producing reports using data in all those different files that seemed to me were logically related and were part of one logical system that needed to have some sort of packaged software product to automatically store and retrieve that data and relate that data together to present it to some other software, like a formatting engine that would print a report and compute numerical calculations.

There were query languages that had been extended into report writers on the mainframes, such as Focus. And I felt that that was what was needed but you needed a very open and flexible architecture in terms of the data being modeled as tables and a database data dictionary that could be queried to find out which tables were there and what their elements were and how those tables could be related together.

And then you needed a very flexible software package that would allow you to relate anything to anything else. Just as if these things were stacks of index cards sitting on your kitchen table. Because the databases that I did buy and install and use on behalf of Bechtel, even though they did save Bechtel probably 90 percent of the cost and half the time of building the applications, it was clear to me that they still weren't as flexible as they needed to be.

You could have one table such as customer that was master of orders and orders was master of shipments but that meant that orders could not be the master of any other table and I thought, well, why? Why is that restriction there? And it was really just because of the physical layout of the data. How pointers could be arranged for maximizing the performance across the disc drives.

It seemed to me that disk drives were going to get bigger and faster and processors were going to get faster and faster but humans weren't going to get any cheaper and what we really needed to do was make software more flexible. Even though it might be slow at first but the hardware would speed up the software and the flexibility and power of the software would make it invaluable.

So, it was during that experience that I decided that my career was going to be building information management software products and that the products I would build would be as easy for people to use as possible. Even though it might inconvenience the computer.

And with that I knew the best opportunities for creating software products were at the microcomputer companies, the new industry of microcomputers.

My dad and I co-authored a book about programmable calculators and another one about microcomputers so I knew that there was this new industry, Apple with one architecture and other companies that were based on the Intel 8080 architecture.

And there was a family of companies that all used the same operating system and I thought, "Well, that's probably a good idea. If you write an application once, then it could theoretically run on all these different things." And I thought well it was an idea that was going to work industry-wide.

So I went down to Silicon Valley to interview. I interviewed with Steve Jobs at Apple and I interviewed with Roger Melen at Cromemco. Steve noticed that I was wearing a tie to the meeting because I had come from the city where I was working at Bechtel. And when you worked on database systems that checked the parts at a nuclear power plant in conformance with the Nuclear Regulatory Commission requirements, you wear a tie. ____________ might call you into his office to write a memo about how the system's going to conform, otherwise the Nuclear Regulatory Commission's going to shut down construction of their power plant. So it was a very official, straight-laced work environment.

I came straight from there to my interview with Steve. And he had hair down to his waist and wrinkled clothes and asked me if I knew 6502 assembly language. He made some comment about the suit right off the bat and then his first question was if I knew 6502 assembly language.

And I said, “No, but I went to Berkeley and we learned the theory of assembly languages and if you’ve seen one, you’ve seen them all.”

He sort of interrupted me and said, “Don’t wave your diploma at me.”

And I thought, “Well, that’s strike two.” And then I thought, well, my only chance is to make an appeal that it’s a good thing that I know about the business side of the world because, Steve, I think these little computers are going to be used for lots more than keeping track of recipes or playing video games. I think they’re going to be used for keeping track of all sorts of business data. And word processors and company lists and customer lists and mailing lists and there’s going to be integration between those word processors and those lists of people.

And he just said, “I don’t think you’ve got the right approach for what we’re trying to revolutionize here at Apple.” And that was the end of the interview.

Now, six months later, after the venture capitalist came in and put a business plan together…

**Johnson:** Do you know what year this was?

**Sippl:** About ‘77?

**Johnson:** Because it’s interesting to fit that into Apple’s timeframe, too.

**Sippl:** Yeah. I think maybe there were 40 people working there.

Six months later they were looking for exactly me. But at that time Steve was looking for 6502 assembly language guys who probably drew better graphics so that the video games ran better or something like that. So for the job he was hiring for, maybe I wasn’t the guy.

## Cromemco And Developing The Design For A Relational Database

So I went over to Cromemco which was one of the bigger companies at the time. They had figured out that they could sell a $3,000 microcomputer to replace a $9,000 Wang word processor.

**Johnson:** Ok. I was trying to fit Cromemco into the picture…

**Sippl:** Yeah. They were an exciting company for a while. They could have been Compaq. They could have been Sun. They were way ahead of both those companies.

**Johnson:** Were they focused on that Wang word processor marketplace?

**Sippl:** Well, that was one market. I just bring that up because they had realized that microcomputers could be used for business purposes. They had targeted the business use market as opposed to home use. For all sorts of business applications. They did have graphics cards but they were high-end graphics cards that were used by weathermen and TV stations. Also industrial applications and business applications for microcomputers.

So I thought maybe I'm a little more at home here and I took the job there.

**Johnson:** Whatever happened to Cromemco?

**Sippl:** They rapidly went from like $4 million to $20 million to $40 million and about when I left it sort of leveled off and went back toward $4 million.

**Johnson:** Did they get bought out?

**Sippl:** No, I think they sold their assets at the end of the game but I don't think it brought very much. Which was too bad because they were probably the highest revenue microcomputer in about '78, '79. They were probably leading the microcomputer market.

But when the IBM PC came out in '81, they had not completed developing their super microcomputer; they could have had the first UNIX super microcomputer. Roy Harrington who was there wrote the Cromix operating system, which was a UNIX look-alike operating system, and he was one of my guys.

He was at Cromemco and he built the operating system that ran C/PM applications. That was the predecessor operating system to MS-DOS.

So Cromemco sold a look-alike of the C/PM operating system and Roy wrote a UNIX look-alike operating system because Cromemco was too cheap to license UNIX from AT&T. They got AT&T to agree to license it and that was a miracle in itself but AT&T wanted $100 a copy royalty or something. Cromemco was accustomed to putting the operating system on the boot track of every disk. So if they sold the word processor it came on one floppy disk and the boot track was the operating system. And they thought, "Well, how can we pay $100 for every piece of software that we ship if we want to put the operating system on the boot track."

I was trying to tell these guys, "No, you got to stop thinking that way. You're talking about a multiuser machine, you're going to start selling a super micro server. You've got to start thinking that the operating system's going to come on ten disks, and it's going to come with a manual and training classes."

They were like, “Oh, no. I don’t think we want that.” So they had Roy write a scaled down version of that operating system that they could distribute freely. That was Cromix and it was the first UNIX-like operating system on anything other than the PDP-11s. It was sort of a breakthrough product.

So I had spent two years building a report writer for a primitive database system that Cromemco had and designing a complete multiuser relational database with a forms package, a report writer and a C programmatic interface that I called CRIS – the Cromemco Relational Information System.

We shipped the report writer which was DBR, Database Reporter, and that was the first report writer for a microcomputer system. Also the first software product ever written in anything other than assembly language.

**Johnson:** What was it written in?

**Sippl:** It was written in C, and I also used YACC and LEX.

**Johnson:** I never heard of either of those.

**Sippl:** They were important parts of the UNIX operating system at Bell Labs. If you were writing a C compiler or a Basic compiler you would use YACC, which is an acronym for Yet Another Compiler Compiler. And LEX is short for lexical analyzer.

They were quite well known in the UNIX world. Most computer science students at universities were using them for student projects. LEX is what I used for the scanner to break down my report writing language characters into tokens like key words and various names. Then the tokens are fed to YACC which knows the grammar, like what order the tokens should come in and what token should generate a statement. One of the two authors of LEX when he was at Bell Labs is Eric Schmidt, who is now the CEO of Google.

But so anyway I built the first microcomputer software product written not in assembly but in C and YACC and LEX.

**Johnson:** Was running on this Cromix operating system?

**Sippl:** It actually ran on both the C-DOS which was their knock-off of C/PM as well as Cromix.

**Johnson:** Okay. But it was on the Cromemco platform and therefore it was going to those customers that used their microcomputers.

**Sippl:** That's right.

I had written three documents totaling probably 200 pages of designs for this Cromemco Relational Information System that was going to have an even more sophisticated report writer that dealt with even a more sophisticated database and the database was going to be a multitable, multiuser relational database using B-trees for indexing and record locking for ________ control. It was going to be a true minicomputer/mainframe developer database management system. With a forms package, query language report writer and C programmatic interface.

**Johnson:** At this time, were you trying to conform to the standards that were being developed for relational databases?

**Sippl:** I was unaware that those standards were being developed. I wrote most of these documents and then discovered Ted Codd's papers.

And I was shocked that IBM let him publish those papers. I couldn't believe it. And I think even C.J.'s first book might have been out around then. I got hold of one of the early copies of C.J. Date's *Introduction to Database Management Systems*, Volume I, First Edition that described Ted's invention and described his papers. Because Ted's papers were pretty hard to read.

But if you got Chris Date's explanation, then you could understand the relational algebra, the relational calculus and the upcoming SQL theory and transaction language. However, I had already designed my system and just coincidentally did the exact same thing – allowed you to join tables together arbitrarily based on…

**Johnson:** That was a remarkable coincidence.

**Sippl:** It was and I'm sure this happens all the time. Multiple inventors invent things and are unaware of each other just because nature has certain models, certain ways, and if you're going to make progress you're going to have 99 guys with a bad idea and one out of 100 guys has the right idea. And if you've got 500 guys trying, you'll have five guys that have the same right idea.

Sometimes I like to think my database was contemporaneously invented with Ted's theory but actually Ted's documents way predated mine. I think his seminal paper was in the very early '70s.

**Johnson:** It was in 1970 but because it was so difficult to understand his ideas didn't get broadly used for several years.

**Sippl:** It was difficult to parse and understand that what he was talking about was the same system I worked on at Bechtel about parts and the submittals and the rejections and the re-submittals and acceptances.

In order to realize that his mathematical models were talking about data processing – replacing COBOL ISAM tables with one data dictionary model multitable and multiuser database management system with a report writer, that was a bit of a leap.

And so it was interesting that I had—I think the value of all my design documents wasn't the strength of the relational model as much as I knew what to do with it. So when I saw Ted's designs through Chris's explanations, I realized that someone had already discovered the basic relational model and described it and had defined this relational algebra and this relational calculus.

I looked at that and I asked myself should I reinvent my design to conform to this exact model in all its fullness – mine was sort of a subset of the model, a useful subset but still a subset – or should I go ahead and implement what I designed.

You have to understand at the time Cromix was designed for a Motorola 68000-based machine which was more of a minicomputer architecture. But the initial implementation still ran on an 8-bit Z80 based processor. And Cromemco was late getting out their 68000 processor for it. It was going to be another year away.

So I thought I better try to implement this for the processors that are here today. And I guess when I realized how late the 68000 was going to be, I realized that I really have to ask the founders to let me leave Cromemco and try to implement these designs on a more powerful computer, a wider word, more __________ address, more memory. Because we could only address – you're not going to believe this – 64K bytes and that was the total address space for the Z80.

With Cromix you could add multiple banks of 64K and sort of back switch but still any one process could only run in 64K. The Onyx computer had just been developed by Bob Marsh and Tim Myers and they had succeeded in licensing UNIX from AT&T. That machine was going to sell for around $10,000 and have a Z8000 in it instead of Z80. And they were going to sell a machine that had a quarter megabyte of memory and another one that had a half a megabyte of memory. And that had a 10 megabyte disk drive. Hard disk drive. Not a quarter megabyte floppy disk but 10 whole megabytes right there.

If I didn't beef up my design to be SQL, I could run on a quarter megabyte or a half megabyte multiuser machine with a 10 megabyte disk drive and it would work. It would run fast, it would support four unique users simultaneously and it would work. So I decided not to change my design but to implement what I had.  And that became the basis for Informix and I did get permission from the founders of Cromenco to market it.

**Johnson:**  They were willing to license it to you?

**Sippl:**  They were marvelous people really.  Harry Garland and Roger Melen.  They were two Stanford engineering professors that started that company.  And they really ran it like a graduate school.  If they hired you into their R&D Department, they gave you an office and a computer and asked you what you wanted to do.  I asked if we had staff meetings or anything.  And they said, “No, no.  Just, you know, after six months if you got something going, set up a little presentation.”  So to them it was sort of like being a Master’s student and you have to show that you’re making progress on your research to your dissertation committee.

They brought that academic model to Silicon Valley – which is probably what’s going on at Google right now – and it was great.

So I did produce their report writer product and that was a great product.  And I produced those designs and, at the end of the day, they said, “Sure, go off and implement this if you think that you can get this built.  Just let us have the right to be your first OEM and go for it.”  I think they figured I was going to go out and raise a little bit of money, spend it all, go broke and come back and bring back a lot of technology with me.  Hey, I think this will work.  Have you guys got that Motorola machine done yet?

I don’t know exactly what they were thinking but they were probably also getting tired of me telling them about this tremendous revolution I saw for database management systems to run on multiuser computers other than mainframes.  Running on machines probably running the UNIX operating system.  And this relational model was going to change the face of data processing.

They were two hardware guys.  They were fire-in-the-valley types, you know, start a company selling circuit boards and bags of chips.  You soldered the chips into the board and you put the boards into the mother board and you hook up the CRT to it and you’ve got a computer.  And that was their background.  They sort of felt software should be free.  They saw it as a kind of overhead.  You had to have a lot of software with a machine or no one would buy the machine.

And I said that I really think that software is going to be an industry and this relational database thing is really a revolution in database architecture.  It’s going to be the most popular software product of all time.  Everyone needs one.  Every business is going to have to buy one.  And they kind of thought I was nuts.  They weren’t quite sure Microsoft had a real model either.

At the time Microsoft would create a basic interpreter product and license that basic language product on an OEM basis to companies like Cromemco. Cromemco said, “Well, that’s nice but, you know, you want 100 grand for a paid up license for that basic product. We’re going to go and pay 50 grand to this other company and they’re going to do just as good a Basic. Because they can look at yours and see what you did and they can program it so we’re going to be smarter and we’re going to pay just 50 grand.”

They hadn’t really grasped the fact that software was going to be mass-produced at some price point and some value proposition. They really weren’t there. So I think that they just really didn’t understand my ranting and raving. They were probably about done with me not producing more products but talking about this grand revolution.

## Founding Relational Database Systems

So I went back to one of the young ladies that was kind enough to date me when I was running that cancer class and I sold her 10 percent of the company for $20,000. She was my ex-girlfriend at the time and I didn’t even know she cared for me. But she did give me $20,000 and with that I started Relational Database Systems (RDS).

**Johnson:** In 1980?

**Sippl:** 1980, right. I think it was around February. And we started building that product. The first pieces we did were the access method and the report writer.

**Johnson:** Who’s we?

**Sippl:** I hired a fellow by the name of Bill Hedge who I had gone to high school with. He was a music major turned computer programmer in order to pay the rent. And Laura King who had worked with me at Bechtel as a contractor. She was also in my department at Cromemco. She actually got the job at Cromemco because her boyfriend got her a job there.

**Johnson:** She left Cromemco to come to work with you?

**Sippl:** Yeah. Originally she left Bechtel to go to work for Cromemco because her boyfriend got her a full time job there. Then she called me up and said, “Hey, you ought to come interview here.” So that is why I went to interview there.

I got hired to be her boss and then I left to start the company. She later joined me. Bill Hedge joined me as one of my first programmers. Bob Macdonald joined me as the first sales guy, marketing guy. A guy named Neil Blumenfield joined us as our first customer support person. And Roy and then his wife Ginny Harrington.

So, bringing Roy over from Cromemco, that was the major coup. He was really the world-class computer scientist of the group. He had been a Multix systems programmer at MIT and so he had written operating systems. He helped write Multix. He wrote Cromix from scratch. He was just a world-class genius in computer science.

Before he came to work for Cromemco he had worked at Watkins Johnson and they had allowed him to go to Stanford and get his Master's degree while he was working there. So he had come to Cromemco from Stanford with all the latest software technology and thinking. Like networks – I think he took the networking class from Bob Metcalfe, the inventor of the Ethernet. When he came he started rewriting all the code I wrote, which was fine.

**Johnson:** Were you running the company at this point?

**Sippl:** Oh, yeah. I was the founding CEO and I was the CEO all the way until late in '99. So we built the report writer and the access method that was going to be our B-tree access method. We sold that as a separate product called C-ISAM so that was the first indexed file system for C programmers for UNIX and that was the underpinnings of our relational database management system.

We put a database data dictionary on top of those ISAM files so that the report writer could allow you to use field names and table names. And then we put the forms package on top of that. And then the query language that was used in the report writer and the ad hoc query language informer was a truly relational query language and report writer. It was based more on the relational algebra than the relational calculus. SQL is more based on relational calculus. But the calculus is harder to implement. It takes a lot more memory to run.

Back then, in '80 to '82, people were still debating whether a relational database could be run on anything smaller than the world's largest mainframe. There were stories that IBM in Santa Theresa was trying to ship a product called DB2 but it was bringing the whole machine to its knees whenever it tried to run queries, so they couldn't ship it. So the relational model was getting a bad rap as being a pipe dream, something that was infinitely flexible but infinitely resource intensive and would never be of use to anybody but the CIA.

I made the pitch that if you based it on the simpler relational model – the algebraic model – it didn't need quite so many resources. It wasn't quite as automatic and not procedural but it was really powerful anyway. And I convinced enough people of that that we sold lots of C-ISAMs and lots of report writers and when we had the full Informix database management system we sold lots of that.

**Johnson:** So the report writer was called Informer?

**Sippl:** That was the query writer. The report writer was Ace.

**Johnson:** OK, so when you packaged it all together you called it Informix?

**Sippl:** We called it the Informix Relational Database Management System. It had the forms package for building input/output interactive screens – that was called Perform, the Informer query language, the report writer that included the Informer query language and our C-API.

It was not based on SQL but it sold like hot cakes.

**Johnson:** What platform were you running this on?

**Sippl:** The Onyx computer. Of that 20 grand, we spent 10 grand of it on the Onyx computer. And opened offices. About five offices. And telephones where I got the hold button with three lines. And my ex-girlfriend came in and said, "Why does everyone have a stapler? Not everyone needs a stapler. What the hell are you going to do with these three phone lines?"

And I said, "Well, I actually hope this whole thing is going to sell pretty well." We had to raise some more money so we had to sell stock very cheaply. I sold stock to the lawyer that incorporated us – Al Mark – and he brought in a business consultant client of his, Dan Sulley. And Dan advised us on some sales strategies and he ended getting so enthralled with the product that he bought stock. And a real estate developer that was a client of Al's, Jay Benedetti, bought stock. And so that was my board of directors.

We raised about $183,000 of initial startup money. And that got us to a quarter million dollar revenue one year, half a million revenue the next, two million the next, five million the next. I think it was when we were on the way $10 million a year when we had several OEM contracts including one with Altos. Kapil Nanda, VP of Engineering at Altos, realized that we were a hot company and they were OEMing our product and Oracle. And he realized that even though our product didn't run SQL, it sold better because it was less expensive, easier to use, easier to install and ran better on the smaller, multiuser machines which is what they sold lots of. He was selling our database system by the thousands whereas every now and then he'd sell a copy of Oracle.

**Johnson:** How long did the development process take after you started in 1980?

## DECISION TO INCORPORATE SQL AND DEVELOPMENT OF INFORMIX 4GL

**Sippl:** We were shipping C-ISAM and the report writer within a year or two but I think it was probably ‘83 when we started shipping the entire database system. By ‘85 we were on Version 3. And we had also created a new product that was sort of a combination of our forms package and report writer. Kapil convinced us that we needed to switch to SQL. Kapil and the guy at Siemens that was in charge of our OEM relationship, another computer science genius, Hans Strauch- Zimmermann. He ended up running all of the computer division for the engineering and product management of all the computers done by Siemens.

Hans and Kapil separately convinced me that we needed to put SQL in the product. So around ‘84 or ‘85 we put in SQL and then we came out with this consolidated product which was Informix 4GL which was one language with if/else statements and while loops that sewed together the control flow to put up your forms, to run your reports, to do database transactions. And that was the first SQL fourth generation language in the industry.

**Johnson:** Was it the fact that the hardware had gotten more powerful in the interim that made it possible to do the SQL?

**Sippl:** Yeah. That was part of it but really just the evolution of our thinking about database management systems. Really the question was is there an opportunity to replace COBOL as the default data processing package.

COBOL with database statements interspersed in within it – embedded SQL for COBOL – was still hard. It was still hard to create database applications. And our customers were telling us, “With your forms package I can whip together a form in half an hour. With your report writer I can create ten powerful reports in an afternoon. With COBOL those two projects would take me months. However, with COBOL, I can create an application that puts up a form, adds, deletes and updates records, finds a bunch of records and prints a report on them. So, I can use the if and else statements, and while loops, and sub-routine calls, and other variables of COBOL, but I’m not very productive.”

I had one VAR company, MBDC, that used our database system in systems they sold to universities. A guy named Chris Maloney there was a big advocate and he said, “Look, I’ve got these 30 programmers working for me. They’re not very good COBOL programmers. If we could invent a programming language that was as easy as your forms package and your report writer in SQL but still have the flexibility of if/else and while loops, we could turn these guys into very productive database applications programmers.”

So Chris and I, even before he worked for us, designed Informix 4th GL. And right around then we brought in a guy named Steve Goldsworthy to run our engineering group and bring more process and more management to our engineering group. Roy was clearly a world-class architect at software, an implementation genius, but managing 20, 50 people, wasn’t really what he wanted to do. He didn’t want to give up technology to go into management.

So we hired Steve Goldsworthy who had grown up in the management ranks of Hewlett Packard and had done some entrepreneurial companies including Sorcim, one of the first spreadsheet products. Steve joined us as our VP of Engineering and Steve and Roy as a team implemented Informix 4GL and I am sure we sold over a million dollars of that product. That product helped put us on the map.

**Johnson:** You keep talking about OEMs. Was that your primary way of marketing? You also mentioned VARs.

**Sippl:** No. We had three channels: end-users, VARs and OEMs. And that was another reason for our success.

Oracle was mostly selling to end-users. Because their initial implementation was on the VAX with the VMS operating system. That gave them entry into the large Fortune 500 companies. Whereas we were targeted for the super microcomputers running UNIX. And I wanted to be *the* database management system for UNIX. And grow with the UNIX market.

Whereas Larry [Ellison] wanted to be *the* database management system for business. Period. And so he was going after mainframe ports and inevitably he was going to port to UNIX. In fact, when Informix was about to go broke at one point in either ‘81 or ’82, there was an ad from Oracle in the help wanted section looking for somebody to run their UNIX database group. So I went over and interviewed because I thought the game was over. It was getting to where I was putting the payroll on my VISA card. And my ex-girlfriend didn’t want to come up with any more money.

So I went and interviewed and Bob Miner and I hit it off great. He really wanted to hire me and I kind of wanted the job. He sort of convinced me that this would be an easier way to go through life. And I went and interviewed with Larry and I think he and I even got along reasonably well. But at the end of the interview he asked me, “Which would you rather do? Come to work for Oracle or be running your own company?”

And I said, “Well, if I had the money I would run my own company. But I don’t have the money. So coming to work here and pioneering relational databases here makes perfect sense, too. You guys have a strategy that you want to start and get going and I can certainly do that for you.”

He didn’t hire me. And I think he must have thought that if someone laid a bunch of capital on me I would leave again. I don’t think I would have, actually. And I think I would have done fine at Apple, too, for that matter. But there are people that love to debate that. They say that I cannot be managed, so there is that element.

But, I didn't get the job so I had to go back to building Informix and I think I did manage to get more investment from the lawyers and from the guys on my board.

Anyway, Informix 4GL was really the major thing that differentiated us from Oracle and Ingres. Those guys finally did have versions that ran well on UNIX on the operating systems that ran on machines smaller than, you know, a half-million dollar machine. When the small machines finally grew up to have 2 megabytes of RAM and 40 megabytes of disc, where they could run Ingres or Oracle, then we had competition but by then I had Informix 4GL.

It was probably in '84 that we shipped Informix 4th GL. By '85, '86 it had hit its stride. So we went public in '86. We had done $10 million in revenue and we were almost done doing our $20 million a year in revenue. So we had succeeded in doubling revenue every year for five years. And we had been profitable for it must have been 18 or 20 consecutive quarters.

And it was all out of necessity because although Kapil Nanda got Dave Jackson at Altos to invest $2 million in 1984, we never really got any venture capital, as many meetings as I had, as hard as I tried. We were always seen as the innovator on UNIX but we also ran one eighth the size of Oracle. Oracle was doubling every year so even if we doubled every year we would never keep pace with them.

Ingres got venture capital in '80 and in '84 Sybase did, but I never succeeded. But I did succeed in doubling every year. And I did succeed in keeping pace with Oracle all the way until about 1989 or '90. We took it public in '86 and then we continued to grow.

Right around '88 or '89, I got this new idea that databases should hold more than just fixed length records with fixed length fields. That you should be able to hold what we called binary large objects or blobs. So we invented the blob and it was because one of our VARs wanted to store images. Wanted the Informix 4GL to build applications to do management of digital images. Bob Marsh , the same guy who founded Onyx, was heading up that VAR. So we came up with a version of 4GL to manipulate digital images and we put the feature into our database engines to hold arbitrarily large fields.

I bought a free text search engine that I was going to incorporate with the database engine to allow free text searching on those blobs that had text in them. And then I realized that really we should put spreadsheets and word processing documents and every kind of office automation data ought to go into the database as well so you can search for it and find it.

And I thought that the file system of UNIX – all those files – should actually be stored in an indexable database as well. That you should be able to search for those files based on attributes such as the date the file was created and who created it. And what is in the file using the free text search feature. And you should be able to add more fields to this record if you wanted to keep track of more than the date created and who created it and what the permissions were, that’s what databases are for – creating more arbitrary fields that you put data in and search on.

I started into this notion that we should have an alternate file system that was an indexed file both indexed in terms of SQL style B-tree indexes as well as search engines. But I needed office automation products to store and retrieve into that engine and I didn’t think I would be able to talk Microsoft into it. Because they had gone and cut a deal with Ashton-Tate and Sybase to bring an SQL database system to OS2, I think it was.

## Merger with Innovative Software

So I went and bought a company that had word processors and spreadsheets as products and that was Innovative Software in Kansas City. And that turned out to be financially a major disaster.

Although the technical vision was absolutely correct. As you can see 17 years later, people are still creating a lot of excitement about being able to search files on your operating system. What I was trying to do in ‘88 was what eventually became the content management industry. Which Documentum helped pioneer. And in fact I think we’ll see upcoming versions of – what Bill Gates has been talking about for the next version of Windows for the last three or four years is that instead of just searching based on the name of the file name or part of the file name, you will be able to search based on content. That’s what we’re going to see and that’s what Google desktop search does and all this stuff is going to happen.

**Johnson:** You were just ahead of your time.

**Sippl:** I was way ahead of my time but I could have been right on time if I would have just executed that better. If I could have just come up with other ways for those products, Microsoft Word and Lotus 1,2,3, to store and retrieve into and out of the database. Which I probably could have done with the right marketing campaign and right partnership relationship building, somehow I could have probably gotten that done.

But instead I acquired this company and we didn’t meet performance expectations financially after the acquisition and the stock fell quite a bit and the company went into a little bit of chaos. I had taken my eye off the major database industry market and I put a lot of the guys from Innovative Software in charge of the company. They didn’t understand the database market, the enterprise software market and so it just wasn’t working. So we brought in Phil White to change all that. Get us back to our database enterprise software roots.

[Ed. Note: The interview on 12/13/2005 was suspended at this point and was continued on 01/12/2006.]

**Johnson:** I've put together a timeline based on the list of major events which you sent to me and the information from our previous conversation. I'd like to go back to the discussion about the products that were integrated to become Informix – the report writer, the forms packages, etc. – and get clarification on what all was integrated together to become the product that was called Informix.

**Sippl:** We had these things priced separately but hardly anybody ever bought them separately. The entire package consisted of the query language Informer, the PERFORM forms package, the ACE report writer, and another component which was the C programmers interface. That was called ALL I think it was. Application something Library.

**Johnson:** Okay.

**Sippl:** That was similar to C-ISAM except it used the database data dictionary whereas C-ISAM was just a retrieval method where you didn't have to create tables with names and with columns with names types. But the relational database system – there was also a program called DB Build that you used to create – I can't remember whether it was part of this original system or not but that's what we used to create the table and give names to the columns, names and types to the columns.

The application programmer interface for Informix, prior to Informix SQL, the non-SQL version of Informix, used a layer on top of C-ISAM that allowed you to use table names and column names. That programmatic interface was not itself relational. The relational things were the query language and the report writer.

So it wasn't until '84 that Hans Strauch-Zimmermann of Siemens and Kapil Nanda of Altos, our two major OEMs, convinced us to do an SQL version of Informix. And so we did that and that's when we added embedded SQL for C to our product line. And embedded SQL for C replaced this application program interface for C programmers, this ALL.

And then inevitably we also did an embedded SQL for COBOL. But the most important thing we did was our own fourth generation language, Informix 4GL.

Informix 4GL was the consolidation of the SQL retrieval and transaction language syntax with our forms language syntax and our report writing syntax. And it was that consolidation of those three chunks of programming capability that was the first SQL fourth generation language. And that's what made the company, frankly.

Because our early SQL database engines were not as robust as needed in terms of transaction capabilities.  So our first SQL products did not have commit and rollback and audit trails.  Whereas our competitors, Ingres and Oracle, did.

But we got away with that for a few years because of the strength of Informix 4GL.  Informix 4GL made it so easy for people to develop applications that people would buy our database engine because that's the database engine that worked with the fourth generation language.

Similar to Progress at the time which had a very strong fourth generation language but did not really have a relational database system underneath it.

**Johnson:**  Was that the company name – Progress – or was that the product?

**Sippl:**  It was the product before they went public.  Inevitably they changed the name to Progress.  Much like all of the companies in the database industry started out as something else and inevitably changed their name to be the name of the product.  Oracle started out as Relational Software, Inc.  Ingres started off as Relational Technology, Inc.  And Informix was Relational Database Systems Inc., RDS.

I used to get Oracle's faxes and emails and paper mails.  We got a fax from IBM once about a contract that they were doing.  I thought, "Why aren't we doing this contract with IBM?  We're the right company.  Got the wrong company name here on the contract, that's the problem."

But it was that fourth generation language that was the strength of the company during the mid-80s.  Informix OnLine came out in the late '80s.  That was the ultimate transactional high performance SQL database engine for the UNIX operating system.  And that was the strength of the company for the late '80s and early '90s, from about '89 to '95.  That pretty much carried the company.

That was the engine that inevitably became multiprocessor capable and all of the super high performance features that allowed us to beat out Sybase and take second place in the relational database industry behind Oracle.  Which we were never going to catch because they started off eight times bigger than we were.  We were doubling every year but so were they.

**Johnson:**  I'd like look at the list of questions I sent you previously to see where we are on the timeline.  What I have here is up through '86, '88 and you just said that in the late '80s Informix OnLine really became the driver.

**Sippl:**  Yeah.  Toward the end of the '80s.

**Johnson:** And you mentioned the last time, too, that about 1988 you began to incorporate the ability to handle other than just fixed length fields.

**Sippl:** Right. Blobs. Binary large objects. I believe we were the first to implement the ability to have a very large field in a database. Intended to hold office automation type documents. And, in fact, I acquired a free text search product with the intention of incorporating that into the SQL language so that you could search for records and then within large documents and in fields of records you'd be able to further narrow down your search using free text search type syntax. But we never pulled that off.

**Problems After The Merger With Innovative**

I acquired Innovative Software because they had the Smart Office Automation Suite and they were coming out with a new fancy spreadsheet called WingZ. It was promoted with a wonderful marketing campaign. The only problem was the product was a year and a half late and every month our marketing department spent another – I don't remember what it was – half million, two million dollars, some outrageous amount marketing it. It just never shipped and it really started killing the company.

There were cartoons in *Computer World* about WingZ being the most famous line of bags. We would give out WingZ bags at all the trade shows. At the first trade show we had an orange one. The second trade show we had a red one. The third trade show, when we announced the IBM version, we had blue one. Everyone had one of these WingZ bags and the *Computer World* cartoon was about it being the most popular software luggage in the industry.

But it demo'd great. Three-dimensional graphics. It would build pie charts and bar charts and let you spin your bar charts around three dimensional bar charts in the sky and the marketing campaign was gorgeous. We had Leonard Nimoy to narrate it. We built a little spaceship trade show booth. Oh, unbelievable. Expensive as hell. But it was killing the company because it didn't ship and then the Smart Suite revenue started to fall.

So, at that point I knew that I needed some better management in the company other than me or in addition to me. I knew I needed something because the management team that I put together as part of the merger wasn't working. We were spending a lot of money on the office automation software. We weren't getting the integration between the office automation and the database products done. And we were taking our focus off our strength which was the database systems.

So it was a good idea abstractly to try to merge the contents of database search with the elements of office automation documents such as word processing documents, spreadsheets, emails, but…. Which is now being done. I tried to do it in the late '80s but it was just stretching too far.  And one of the reasons I was trying to do it was because it was an innovative idea and I felt everyone should be able to find their office automation, find their budget spreadsheet, or their email they sent last week, just as easily as they could find a customer's record.

They could find a customer's record, they could find his order, they could find out whether his order shipped.  But they couldn't find the contract.  And I felt, "Why isn't the contract in the database?"  So they could find it like they could find anything else in the database.  It was a good idea abstractly and it was an opportunity for us to really differentiate ourselves from Oracle.

The Wall Street analysts had convinced me that since Oracle, being eight times our size, had eight times the R&D budget, eight times the marketing budget, how were we ever going to compete.  I really shouldn't have listened to them.  Because we were doing fine. But inevitably they wore me down and I felt we needed that sort of differentiation to try to break away from this eightfold size difference.  But it just wasn't working.

It was a good idea.  A lot of people tried really hard.  It just wasn't working.

We interviewed at first for presidents with the idea that I would remain CEO and we would bring in a president.  Which I think would have worked fine.  We offered the job to a guy named Bob Finnocchio.  And he turned us down and took a job as a VP at 3Com instead.

It's really too bad because sometimes I think that if Bob had taken that job he and I would still be working at Informix and it would be a big company.  We would have probably have fulfilled the vision of allowing the users to search on all types of information.  Because I really felt that the file system of the operating system was – when Thompson came up with the hierarchical file system in the 70s, I thought it was very elegant.  But when it comes time to search for something in it, it's not indexed and is not very elegant.  So I really thought that the UNIX file system needed to be replaced with a database management system that kept everything in it. Kept all the files as well as all of the structured database variants.

I actually had some guys at AT&T pondering this notion for a while and I was going to try and do an OEM deal with them where our database engine was an alternate file system for the UNIX operating system.

And that was one of the reasons I did the acquisition with the office automation company because I had to know at least there was some office automation tool that could store and retrieve office automation documents into this new file system.

But, in any event, back to the management suite.  Bob didn't take the job in 1990 or whenever it was we offered it to him.  Maybe it was '89.  So we decided to change the search to be a CEO search so we could attract a bigger field of qualified candidates and that's when we hired Phil White to be CEO.

A lot of my board really did not want me to do that, did not want me to not be CEO. In retrospect they were probably right. I should have probably just kept trying to find someone to come in to be the president to help me operationally. Or someone who could share power a little better.

It's always been difficult in the high tech industry to have two people at the top of a company. I mean, it so rarely works at all.

**Johnson:** I've heard a lot of people say that.

**Sippl:** So I'm not sure how this really could have gone but I think there are some successful cases where there's more than one person in the top. It works where some sort of committee makes the decisions and I think a good company could have been formed by someone who's very good operationally and then someone who's got the technical vision.

We needed someone else doing the budget, the media, timelines of when things are supposed to be done, coordinating multiple groups together.

**Johnson:** That's always the vision people have when they think of that kind of a combination, you know, but somehow it just doesn't quite seem to work.

**Sippl:** Yeah. So that would have been nice but that wasn't to be.

I guess Ellison did it from time to time at Oracle. He would bring in presidents and they would happily run the operations. So I think it's been done, it's just the presidents usually aren't high profile.

## Decision To Leave Informix And Subsequent Activities

But in any event, that's not what we pulled off. What we pulled off instead was bringing in a CEO. And I stuck around for two or three years as the Chairman and making speeches and trying to help with product strategy. But before too long Phil really wanted to run all of everything himself and saw me as a threat to his power. Which I think is probably the problem that eventually happens when a founder tries to bring in a hired gun manager. That manager really, you know, wants to be in control of the whole company and feels that he will be ineffective if there's some part of the company outside his control.

So that usually means that he and the founder are not going to see eye to eye. If ever the founder has some other idea, the hired gun manager is going to quash that idea or do a end run around that idea. After a while that gets pretty frustrating. And I had always been getting a lot of business ideas from other entrepreneurs who had seen my success in Informix and wanted my help in getting them off the ground.

One of them was Steve Goldsworthy, who was my VP of Engineering for most of the ‘80s. And so I helped Steve with his business plan and gave him some of the funding to start what was to become Vantive Corporation which used all of the technology that he and I built together.

**Johnson:** Under license from Informix?

**Sippl:** Well, no, actually it was ironic. I should rephrase that. It used all of the types of technology we pioneered but actually – and this really aggravated Phil – Steve chose Sybase to build his first version. He wanted to use stored procedures. He had an architecture that required that the server where the database was actually have a fair amount of business logic. And we had never implemented stored procedures in Informix. I wish I had. We had this very popular fourth generation language. I should have made it so that some subset of that fourth generation language could be executed as a stored procedure.

But I missed that one. And the stored procedure was really the beginning of the three-tiered architecture. The stored procedure was really the thing that allowed people to take business logic out of the client side of client server and put it more on the server. And certainly history has borne out – and even in the mid-90s when I was CEO of Visigenic Software, that is exactly what that company was doing when I took them public. Pioneering the applications server which was the ability to have a lot of business logic on the server so you can have thin clients. Eventually very thin clients when the web browser was invented.

And so he had his engineering reasons but commercially it wasn’t very harmonious with Phil. He and I were on the Board of that company and he ran it and I never really worked there full time but I did help him get it going and we did successfully land venture capital money from Mohr Davidow and some other firms. Bill Davidow was very helpful getting that company off the ground. It eventually went public and grew to have a billion dollar market cap all of its own.

I actually made more money on Vantive than I made on Informix because I didn’t need to sell all the stock the day I went public. I didn’t sell all my Informix stock the day it went public but when Informix went public I was selling every quarter. I didn’t have any money when I grew up so I needed the money. I liquidated most of my Informix holdings long before the order of magnitude value increase.

**Johnson:** So what happened with Vantive?

**Sippl:** Vantive got bought by PeopleSoft. So the database part of Informix is now part of IBM. Vantive is part of PeopleSoft…

**Johnson:** Which is part of Oracle.

**Sippl:** Yeah. True. Part of Oracle. And Visigenic we took public and then it got bought by Borland.

**Johnson:** Oh, it did?

**Sippl:** Yeah. Bought by Borland. So that's my three directly involved successes. I was also an investor in the '90s in lots of other startups. Michael Stonebraker, the founder of Ingres, Gary Morgenthaler and I got Illustra going which got bought by Informix. And that's one of the reasons I believe that the Informix board – some of these venture capital investments I was doing were starting to conflict with Informix's basic products.

In fact, I did not invest in Gain because I thought it would be in conflict with Informix. So I turned down that investment and that ended up getting bought by Sybase for quite a bit of money, I think. Over $100 million. So I said, "Well, I missed a million or two there just out of this conflict. I don't really have the ability to run this company where I'm working so I should leave and just be an investor". And so that's what I did for a little while with Vantive, Red Pepper, Illustra, SupportSoft, TimesTen.

There were so many business plans that I was looking at and investing in that I brought on a partner, Bob Macdonald's wife, Jackie Macdonald. They both worked at Informix, that's where they met. So Jackie and I formed Sippl Macdonald Ventures. Some of these investments were mine on my own before Jackie and I started Sippl Macdonald and some of them, SupportSoft and TimesTen, were Sippl Macdonald investments.

So that's what I did. After Informix, I did small scale venture capital as well as started Visigenic Software in about '93, I think. I did run Visigenic Software so I was on the Vantive Board at the time same time I was running Visigenic Software.

**Johnson:** It's interesting that Illustra went to Informix and then Informix to IBM. Was the Illustra product a big part of what IBM wanted from Informix?

**Sippl:** I don't know. IBM was interested in acquiring Informix before Illustra was even started.

**Johnson:** Oh, they were?

**Sippl:** Yeah. They looked at Informix several times in the late ‘80s and early ‘90s. I think it was the fact that it was the dominant database system for the UNIX operating system. Finally they were conceding that their proprietary operating systems were going to lose to UNIX but it took them a full decade to make that concession.

**Johnson:** So, do you know, are they really focusing on Informix now as opposed to the databases that they developed?

**Sippl:** I doubt it. I’m sure they still have a big installed base in Informix and still sell and support Informix but I believe DB2 is really where their focus is.

## Promoting UNIX

But speaking of UNIX, I think the story of Informix is much more tied to the story of UNIX than any of the other database companies. Because I started the company to be *the* database system for UNIX and that’s not true of Oracle or Ingres or any of the others. We attempted to put our report writer on top of DB2 on the mainframe but that didn’t work out so well.

We were really technology-wise and marketing-wise tied to the UNIX operating system. And one of the reasons Informix succeeded was because we spent a lot of time making UNIX succeed. We joined with Doug Michaels and Bob Marsh and a bunch of other folks who were early pioneers in the UNIX industry to form the UNIX trade association and we started the Uniforum trade show.

There was USENIX but that was a very academically-oriented organization. We started. a separate organization to complement those things and specifically to be a commercial organization. An organization to promote the commercial uses of UNIX.

In the early ‘80s even with Sun’s success, the industry analysts, the Gartners and _________, would never cut UNIX any slack. They would all say it will never be successful as a commercial operating system. It might be good for engineering workstations but, you know, Microsoft dominates the desktop so UNIX on the single user personal computer, that will never happen. And even though Sun is having some headway with engineering workstations, that is just a niche. It will go away when the PC gets more powerful. And UNIX on servers, well, that doesn’t really make much sense either because, you know, there’s the IBM server product and the Digital Equipment PDP-11 and the VAX line. The VAX’s are getting smaller and cheaper. And so basically what we would have is the large companies – IBM and Digital Equipment – spending so much money on the industry analysts that they would literally hold trade shows.

I wish I could find the information about a conference where I spoke because I promised that I would donate it to a museum some day. The conference was called “Exploding the UNIX Myth” and I believe it was run by Gartner. I'm almost positive. They invited me to be the whipping boy for the UNIX operating system. So I was on this panel, ostensibly among people with different points of view, and there was a moderator. And whenever he would fire a question at me it would be along the lines of, you know, who do you beat more, your wife or your daughter? It was a no-win question I was given every time.

**Johnson:** I think it was Gartner that did that.

**Sippl:** You're absolutely right because I think I had a drink with Gideon Gartner after this conference.

**Johnson:** Gideon was still with at Gartner at this point?

**Sippl:** I'm was almost positive it was him. I was drinking a cognac after I just got roasted. And in the middle of that roasting I had to blurt out – I just interrupted the guy's question and I said, "The UNIX operating system is going to be the dominant commercial operating system and you guys are just making absolute fools of yourselves. I'm going to take this binder that says "Exploding the UNIX Myth" with the Gartner logo all over it. And after UNIX is the dominant operating system in commercial computing, I'm going to donate it to some museum to put it under a Plexiglas enclosure."

**Johnson:** You don't know where it is at this point?

**Sippl:** I don't think I have it any more. So we were having this drink and I'm trying to convince him that UNIX is going to be the dominant commercial operating system. That there's nothing inherently wrong with it and in fact, it's as good or better than VMX and CICS. It just hasn't had the databases and the tools to build commercial applications but that's where SQL has done the best and the companies that are providing database tools for the UNIX system are doing really well and more and more hardware companies are putting UNIX on their hardware. Customers want to buy the same operating system no matter what brand of hardware they're buying. So it is going to cover the earth.

And he said something along the lines of "Well, of course".

And I said, "Well, wait a second then." I picked up the binder and I said, "What's with this exploding UNIX myth then?"

And he goes, "Well, it sells papers."

I'm sure he was getting a million a year or more from IBM for inviting a bunch of IBM mainframers to conferences and say that the IBM mainframe is better than a pipsqueak toy operating system.

So, anyway, that was as much of the battle of Informix as the relational databases were.

**Johnson:** See, that's the kind of stuff that's really interesting to come out of these interviews. To understand that there was this need to promote UNIX in order to make Informix part of that marketplace. Create a UNIX marketplace so there was a marketplace for Informix to be in.

**Sippl:** Yeah.

**Johnson:** One of the questions I had for you was about platforms and you've said your platform was UNIX and you stuck with that. So you didn't make any attempt to try other platforms, or did you and it didn't work out?

**Sippl:** The IBM PC worked a bit. We never had anywhere near even a small percent of the market share that dBase had with Aptiva. But when people wanted the same database system on the PC as well as their UNIX machine we did very well and that really helped to sell the UNIX product. That was an equivalent IBM PC product. When they built a database to be a shared database by 1,000 users on minicomputers running UNIX but they also needed people to have mirror images of that database on their PC that might have a selected subset of the data from that database, those contracts we won with almost no competition.

Oracle had a PC version of their product but it was horrible on the PC. It was just too big. And that's one of the reasons we succeeded in the UNIX industry in '83, '84, '85. Because our product was just smaller and sleeker and it took a while for memory to become, you know, big and cheap on these multiuser super microcomputers.

## Support Services Provided By Informix To Customers

**Johnson:** One of the other questions I had was to get a sense of the extent to which the product you were selling involved services as well. To what extent did you have to provide services in order to sell the product? How much was really an off-the-shelf product for your customer base? You had OEM customers so clearly in that case you were selling it to the OEM or licensing it to the OEM. But in terms of going directly to the end users, did it require a lot of support?

**Sippl:** Yeah. It required a reasonable amount of support. Training certainly was one very important thing. And that's how you knew you were about to win a deal – if you got a purchase order for the training for the product. That quite often happened much to my amazement before the customer even started negotiating the price for 1,000 seats for Informix and Informix 4GL. They would say, “Well, we’re not sure that that’s the tool we’re going to use. We want to send some people to training so we can build a prototype of our applications so three people are going to come out next week. Find a spot for the training classes and just send us a bill for the training.” And I thought, “Well, okay. We were about to offer them a 40 percent discount, make it 20 percent.” Because once they’re there for training they’re either in or out.

Once they saw the power of the products, particularly the 4th generation language, very rarely did they not buy the product and so it was important to provide services along the way.

The other thing we liked to do on big projects that were going to take maybe six months or a year to develop – very top-level web systems for customers like the phone company – was to send one or two programmers to the customer and work with their 30 programmers. Those two consultants would be involved in the training of those 30 people and they would be onsite sometimes as long as two or three months working with those people.

Now, of course, I have no idea but I think at Oracle they probably would have sent 30 programmers to be onsite for a year for the same project. They were such a monstrous service organization, but we were just sort of doing what we had to do to support the sale.

The OEM channel was an interesting channel. Most of the OEMs sold to other software companies that produced a vertical solution like a pharmacy account management practice product. And so a lot of those companies didn’t need a whole lot of support. They were sophisticated software engineers and quite often they’d have people with graduate degrees and that have used one or two database systems already in the past. And really all those OEMs tended to do was sell the manuals separately on the price list and the customers would go through the manuals and say, “This is what I’m looking for.”

And sometimes those customers would come and do contracts directly with us and that was always an issue of a channel conflict. Because unlike Oracle we were heavily dependent upon our three channels which were the OEMs, the VAR’s and the end users. But they all complemented each other. We had no marketing budget to get to the end users.

**Johnson:** That’s interesting.

**Sippl:** To speak of in the ‘80s in particular. In the late ‘80s, we were able to do some advertising but we didn’t have venture capital. You know, we just had these small amounts of money that we scraped together so any growth we did had to be from our own positive cash flow.

So, in order to get positive cash flow we would often sell 1,000 copies at a time to an Altos or to a Siemens or some OEM.  With Siemens I think we gave them a very steep discount for several hundred thousand dollars cash up front.  And that would make it so that these companies could distribute lots of copies of the product and they could make money on it.

And then we had the customer support line so that when these customers had questions they would usually call us instead of the OEM.   So it was as much of an advertising strategy as it was a distribution one.

**Johnson:**  So then did you have a customer support contract with these customers?

**Sippl:**   Quite often the OEMs would sell the product but allow the customer support contract to be signed directly with us.  Some of them wanted to have the customer support contract signed with them.  Some of them would sell computers and word processors and database systems and a customer support contract for the whole thing.  So it would be done by, say, an Altos.

But when someone really had a serious database project going, we usually heard from them.  They usually took training classes from us or bought certain support services from us or consulting from us.  And then over the years those VARs that succeeded inevitably decided they wanted to have a relationship directly with us.  So even though we could only afford to have a dozen or two dozen sales people and Oracle had 200, we leveraged the sales forces of all of our OEMs and our VARs.  So VARs like Altos and AT&T, those sales people went out and laid a brochure on a customer saying “Yes, we’re trying to sell you these 100 computers and here’s the database system that it can run with.”  Just getting our brochures out, making those sales calls and getting our presence directly known to the customer through the sales people of our OEMs, that is how we survived without a lot of venture capital.  That’s how we bootstrapped.

And in the late ‘80s, all those customers came back around and wanted to learn more about the product, what new products were coming out.  We have these 12 features we would like to see in the next product.  We’re going to buy another quarter million dollars worth of software this time and we’re going to buy it directly from you.  We want to come out and have a corporate visit and talk about why we want these features.

**Johnson:**  Oh, really?  They went to you when they wanted to upgrade what they were doing with their database.

**Sippl:**   They got our product through an OEM sales channel and it worked and they were going to do the next phase of their project.  Their company was succeeding and this database was critical to our product so they decided we better go talk to these guys and have a relationship directly with them.

So a lot of the customers that we enjoyed in the second half of the ‘80s came from this seeding work that we were sort of forced to do in the early half of the ‘80s because we needed the cash flow from selling through our leveraged channels, less expensive sales channels such as the OEMs. We'd do one deal with an OEM and all of a sudden 50 sales people are handing out our brochure.

To us it was critical but it wasn't as critical to Oracle because they had the VMS products so they could leverage the burgeoning VAX and VMS market and sell a better database system than Digital Equipment was selling and they would win deals. So they would get $100,000 or $200,000 for small deals in that marketplace. We would have to go sell 1,000 copy deals to Altos to get $100,000 or $200,000. So it was a very different game we were playing. It required a lot more low price points and high discounts for channels but if you survived that then all those copies of your product that are out there in the field being used…

**Johnson:** And everybody's got their data embedded in that product.

**Sippl:** And now they need to use the next version and they heard about some hot new fourth generation language or version of your 4GL or embedded SQL for COBOL they've been waiting for and they want to hear about it. And so we survived until the second half of the ‘80s and then flourished because of all that hard work through those risky leveraged channels.

It was tough in the first half of the ‘80s so we never got any respect. Even when we did get an OEM deal with AT&T, they had already done an OEM deal with Ingres and Oracle. But when they ported those products to the early AT&T super micro 3B2 and 3B5, the products from Ingres and Oracle didn't fit. They didn't work. They actually could not even ship them.

So Ingres got a $2 million OEM deal. We got $140,000 OEM deal. And then they proceeded to ship several thousand copies of Informix and not copy one of Ingres. So it was pretty galling, but it worked.

**Johnson:** Do you think that was at least partly because you were on the UNIX platform and the UNIX platform itself didn't have that kind of respect? You would think AT&T would have respect for it.

**Sippl:** Right. Well, I think it was because Oracle and to a smaller degree Ingres started before we did and were better financed. Oracle was semi-bootstrapped as well. I don't know how much money they ever raised. It wasn't a huge amount of money. But Ingres was venture capital backed so they came out of the chute and they could afford to go buy a VAX. I couldn't afford to buy a VAX until three or four years into the company.

So they not only could afford to buy a VAX but they could afford to port code to the VMS operating system and start sales offices in all these cities and have this direct end user sales force.

**Johnson:** That's really, I think, the key to it.

**Sippl:** It was. The VAX, the Digital Equipment products were very popular. The VMS operating system. Ingres and Oracle made their early money off the VMS operating system and that's what gave them the cache with OEMs that they are the relational database for serious business applications because they had all these success stories from large Fortune 500 companies because the Fortune 500 companies bought the VAX and supported VMS. We didn't run on VMS so we were totally dependent on UNIX succeeding versus VMS.

If I had it to do over again, if I could have raised some venture capital, that would have been a good move to run on VMS and UNIX but we couldn't afford it.

**Johnson:** Interesting. That's the kind of stuff that really fascinates me.

**Sippl:** The strategy.

**Johnson:** Yeah. The decision to stick with the UNIX and the unintended consequences because I'm sure you didn't really anticipate that you were going to have to function that way with the leveraging of the sales forces and also that you didn't really anticipate what the positive consequences of that were going to be later on.

**Sippl:** Yeah. Those were pretty much the only channels for UNIX in those days. You have these Silicon Valley companies putting the next Motorola chip into a super micro with the next Shugart disk drive. And porting UNIX to it and throwing it out there and it was not snapped up by, you know, Procter & Gamble. It was snapped up by other value added software companies selling turnkey systems. That was the only real market in the early '80s.

**Johnson:** That's really interesting to understand that factor.

I think that's the end of my list of questions. Once this is transcribed and edited, I'll get a copy to you to review and add additional information if you want. It's been a fascinating interview and I really appreciate your taking the time to do it.

**Sippl:** Thanks. I appreciate your interest in the history of Informix.