An Interview with

LUANNE JOHNSON

OH 385

Conducted by Janet Abbate

on

7 May 2004

Needham, Massachusetts

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

Luanne Johnson Interview

7 May 2004

Oral History 385

Abstract

Luanne Johnson, one of the early women entrepreneurs in the computer software industry, describes how she became a programmer and then how she started Argonaut Information Systems, a provider of packaged accounting software products. She discusses how she simplified the structure and programming of the products for easy maintenance and portability and focused on of-the-shelf sales to medium-sized companies. She talks about her many years as an active participant in ADAPSO and then in leading the ADAPSO Foundation and becoming the President of ADAPSO which was renamed as the Information Technology Association of America. She concludes with stories about her consulting career after leaving ADAPSO.

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
Dan Bricklin and Bob Frankston, interviewed by 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

Luanne Johnson Interview

Janet Abbate: This is Janet Abbate speaking with Luanne Johnson on May 7, 2004 in Needham, Massachusetts. Let's start with a little bit about your background and how you got into computing. You grew up in Orrville, Ohio and your father was a teacher. When you were a kid, did you have any special interest in technology or math?

BACKGROUND

Luanne Johnson: No.

Abbate: Languages?

Johnson: No, none. What I was interested in was writing. I started writing stories and poems very early and then branched into essays and so on. Looking back, I think I would have loved math if I'd been encouraged in it, but it just wasn't something that anybody ever seemed to expect me to do. So I just never pursued that. What I was interested in was writing.

When I was about 11 years old -- I'd been writing stories since I was 5 -- I walked into the room where my mom was playing bridge with her friends. One of the women said, "Luanne, what you going to be when you grow up?" It was unusual to even be asked that question, since everyone assumed that when girls grew up they would be wives and mothers. But I just looked at this nice woman with some disdain and said, "I *am* a writer!" In my mind, there was no "when you grow up" about it. I felt that I was already pursuing my chosen career.

Thinking back on it now, just imagine -- those women must have dissolved in laughter when I left the room! But I took it very seriously at that point.

Abbate: So this would have been the early to mid 1950s?

Johnson: Well, I was born in 1938, so that would have been about 1949 or something like that.

Abbate: So you were planning to be a writer and your dad wanted you to be a teacher?

Johnson: Oh, he wanted me to be a teacher because he'd been a teacher. It was just assumed -- it wasn't even a matter of trying to convince me. It was just -- well, of course you'll be a teacher. I went to college and initially enrolled in education courses, but not with any great desire to be a teacher. But the education courses really turned me off. I didn't like the classes or the teachers or any of it. So after the first semester I switched to liberal arts and created quite a fuss with my father who was very unhappy about that.

Abbate: You were still planning to write?

Johnson: Oh, yes, I was still planning to do that.

Abbate: So while you were in college, you took a trip to California. Where?

Johnson: To Berkeley.

Abbate: For leadership?

Johnson: Yes, I'd been elected president of the university chapter of a combined YM/YWCA. They had a scholarship program to send the incoming chapter president to one of the leadership training schools which the YMCA held each summer. I had the choice of going to one at Union Theological Seminary in New York or one at the Pacific Theological Seminary in Berkeley. I'd been to New York, so I wanted to go to Berkeley. I went out to California for six weeks for that program and, while I was there, I met my first husband and fell in love and decided I wasn't going back to Ohio or the YMCA. I ended up sending that organization about $2 a month for a very long time to pay off the money they had spent to send me to the school.

Abbate: So you just dropped out of school?

Johnson: Just dropped out of school and got married and moved to California. I loved California and Berkeley, and knew that was where I wanted to be.

LEARNING TO BE A PROGRAMMER

Abbate: Tell me how you got interested in programming?

Johnson: It was after I split up with my husband and was raising my daughter on my own. He was supposed to be paying child support, but it never really happened. So I was working full time as a secretary, and raising her and taking classes at the university when I could. It was very hard. I was working in a law office and there was a woman there who had been a secretary for thirty-some years who knew more than most of the attorneys. She was in essence a paralegal, had they had such things in those days.

I found out that after thirty plus years in that job, she was making $20 a month more than I was, and I decided that was not a career path that I wanted to be on. I still wanted to finish college but at the rate I was going, it was going to take a long, long time and I needed to be making more money in the meantime to cover my childcare costs and just to provide us with a decent living. So I was trying to figure out what to do.

I had a friend who was a librarian for the City of Oakland. In those days, you didn't really have people coming out of school trained to become programmers, because no colleges were offering it. So big companies and governmental agencies would offer training to their internal employees to become programmers. My friend had taken advantage of this at the City of Oakland and had become a computer programmer. He kept pushing me to learn programming but I initially thought it was a crazy idea because I didn’t have a math background and didn’t think of myself as the type of person who could do technical work.

Abbate: Had you had any exposure to computers at all?

Johnson: No. But I was pretty desperate. I really wanted to find some way to make more money. So he finally convinced me to take a programming aptitude test and I did very well on it.

Abbate: This was one of those IBM tests?

Johnson: Exactly. Everybody used the aptitude test that had been designed by IBM to determine who would be good at programming, and, somewhat to my surprise, I got a very good score.

Abbate: Was this given by the County government?

Johnson: No, I don’t remember for sure where I took the test. I think it might have been Heald College, which is where I ended up taking the programming classes. I was encouraged by the good score I’d gotten on the test, so I signed up for the course at Heald College, knowing nothing about computers -- except all the bad images that they had in those days.

It was two nights a week, two and a half hours a night, something like that, for several months. And before the first class was over, I was in love with it. I just thought it was wonderful. I loved the structure and the organization of it, and I could just hardly wait to for the next class. I got my certificate from Heald College and really wanted to start programming, but then I still couldn't get a job because people were afraid to hire somebody that hadn't had any actual experience.

Abbate: So the entire programming course was only a couple of months?

Johnson: Yes, I don’t remember for sure how long it took but it certainly wasn’t a year’s commitment or anything like that. It was a big challenge because the course cost several hundred dollars which was very big money to me and I had to pay for babysitting while I attended those evening classes. I had to work overtime to save up the money to do all of this. But I knew right away I'd done the right thing because I really liked it. Then I just kept applying for programming jobs, and answered every ad I saw. Meanwhile I was still a legal secretary because nobody wanted to hire a programmer who didn't have any actual experience.

Abbate: I thought there was so much demand for programmers - somebody who'd been trained would find work readily.

Johnson: Once you had some actual experience, then yes -- then there were all kinds of opportunities for you. But companies tended only to hire people who had already had some experience. Many of them also trained people to be programmers but they gave that opportunity to existing employees, and didn't hire inexperienced people off the street.

Abbate: Would it have mattered if you'd had a college degree?

Johnson: I don't know -- it might have. If I'd had a science or math degree it would probably have made a difference. All I know is I just kept sending in applications and not hearing anything back. I didn't get called for interviews.

WORKING FOR ALAMEDA COUNTY

And then one day I was at the County courthouse. Occasionally as part of my job, if we were late in filing a brief, I would actually have to walk down to the courthouse to get it into the hands of the clerk before the filing deadline. I was there one day doing exactly that, and I saw a notice posted that the County of Alameda was going to hire programmer trainees.

So I put in an application right then and there and went in to take another aptitude test with about 300 other people. I did well on the test again and they called me in for an interview and I ended up being one of three people that got hired as trainees.

It was funny because the other two who were hired were men, and there was an initial six month period when you spent full time in training classes. These two guys complained the whole time about the pay cut they had to take to do this. And I was making $25 a month more as a programmer trainee than I had as a secretary and I was doing something that was fun! So I just felt like I'd fallen into a wonderful situation. It was really great for me.

Abbate: Were there a lot of women in the class at Heald?

Johnson: I was the only one. Most of the programmers at the County of Alameda had been employees in other departments that had been selected to be programmers based on their scores on the aptitude test so there were several women who ended up in programming that way. So once I actually got into programming, I wasn't the only woman, but I was the only woman in that class at Heald out of about 25 people.

Abbate: It sounds as if you had a substantial amount of training from the County.

Johnson: Oh, yes.

Abbate: They sent you off to IBM school?

Johnson: Yes. Back in those days before IBM unbundled, if you were an IBM customer then you could send your employees to IBM training classes. The County stipulated which courses you *had* to take, such as COBOL, but they didn't limit the courses you could take. A lot of the courses were self-paced and I could go through those pretty fast, so I just crammed in every course I could during that six-month training period.

Abbate: So you were learning other languages like Assembler?

Johnson: Oh, yes.

Abbate: And did you ever use them?

Johnson: No, not really. I learned COBOL and Assembler and PL/1 and RPG. I even took a Fortran course which I used. Even after they started to give me some real work, I didn't have enough work to fill up the day. At the beginning, I was only maintaining programs that someone else had written, so I'd figure out the changes that need to be made, sit down at the key punch machine and change the deck of source program cards, and then put it on a shelf by the computer room to be compiled and tested. We never actually went into the computer room. You would put your program on the shelf with a run sheet with instructions for the operators and then wait for them to run the job and give you the output.

Everything had a priority assigned it. Production jobs had the highest priority and some applications had higher priority than others, so if you were a programmer working on one of those, you would get your jobs turned around in three to five hours.

But since I was a trainee, I'd put a job on the shelf and wouldn't get it back until the next day. So, in the morning I would come in and get my output and spend a couple of hours fixing any problems and setting it up to run again. Then I wouldn't have anything else to do until the next morning so I just used all that extra time to take classes.

Abbate: You were working on a traffic court system?

Johnson: Yes. It was just the luck of the draw. As people came through the training program, they would just plug them into the next job opening, whether it was because a department was expanding, or someone needed to be replaced. The job opening that came up when I was getting to the end of the training period was with the traffic court system.

It was a really bad system design because the traffic tickets were actually punched cards and so when a cop wrote a ticket on the top paper copy, a carbon copy of what he wrote would be on the punched card and these would be turned in at the end of the day for an evening shift keypuncher to try to read the handwriting and punch the information into the card. The cards got all bent from being carried around in the cops' back pockets and on a rainy day you could go into the keypunch room and see traffic tickets spread all over the room drying out so that the keypunch operators could get them into the machines without tearing them.

Those tickets had to keypunched and then processed over night to prepare the court calendars for the next morning. It was the only application the County had that had an overnight deadline. The welfare system was run monthly; the election system was twice a year -- everything else had a much longer time frame. I happened to get assigned to the system that had to be processed overnight which meant that I was one of maybe only two programmers out of a hundred who was on call during the night if they had a problem with the program. I was also one of the only ones who was a single mother.

The first couple of times it happened, I called up the babysitter and said I had to go into work and asked if I could bring my daughter over to sleep at her house. After a couple times she said "I really can't do this -- I can't have you calling me in the middle of the night." So then I'd bundle my daughter up and take her to work with me and put her to bed on the couch in the ladies room and prop the door open so I could hear her if she woke up. I kept applying for a lateral transfer. I said, "Please transfer me to some other department because this is the only department where people are expected to come to work in the middle of the night and it's a real hardship for me." But they said "No, our policy is we don't do that".

So for two years I dealt with that, and then I got a call from the man who had taught one of the classes that I took at Heald. He offered me a job at a firm where he had just taken a programming management position. It was a lot more money than I was making at the County and promised me no calls telling me that I had to come to work in the middle of the night, so it took me about two seconds to accept it. In that job, and the other jobs I had later, I often ended up working at night because it was the only time when the computer was available for testing. But that was fine if I knew it in advance because I could make arrangements for my daughter to sleep over with her babysitter or someone else. The problem was getting the call in the middle of the night and having to figure out what to do with my daughter when I was called into work at 3 in the morning. The managers at the County could never get it through their heads that that was a big problem for me.

Abbate: So that was your last government job?

Johnson: Yes, that was the last government job.

Abbate: Were the other jobs any better?

OTHER PROGRAMMING JOBS

Johnson: Oh, sure. The industry jobs paid better, and by that time I had almost two years of experience. That made a big difference. I ended up changing jobs a lot because I was constantly getting better offers for more money and more interesting applications to work on and more responsibility. I would meet people in one job who would move on to something else and they'd recommend me to their new employer and I'd get another offer. I never actually went out and looked for another job. But once you've got some experience and the people that you've worked with start to move around, they know who you are and what you can do.

Abbate: And you moved into these accounting and payroll applications pretty early on?

Johnson: Well, yes. There was a variety. I really didn't start working on payroll until I ended up at CommSci Systems. But they were all accounting-oriented, back-office type of applications. The first job I went to after the County of Alameda was with a freight forwarding firm, National Carloading. I got that job offer because the man who had taught one of the classes I took at Heald College took a programming management job with National Carloading. He called me and asked me to come to work for him and offered me a lot more money than I was making at the County. A freight forwarding firm doesn't have a fleet of trucks. They are more like a broker between people who have freight to be shipped and the trucking companies, so they have the responsibility for the paperwork involved with that.

The system I had responsibility for was the one that kept track of the bills of lading for all the shipments. So it was very heavily accounting-oriented, but I really didn't get into payroll until I got to CommSci. I wasn't doing anything scientific or mathematically oriented -- it was all to do with financial reports of one sort of another.

Abbate: And this was all in COBOL?

Johnson: Yes, all in COBOL.

Abbate: On IBM platforms?

Johnson: All on IBM, yes.

Abbate: What were the models for these programs? Was it kind of an established genre of software?

Johnson: No, it was all custom built. It was a matter of sitting down with the user department and they would tell you what they wanted -- you know, I need a report that looks like this. And then you'd have to work your way back to where the information came in, and build a system that captured that information from all its various sources and produce the report they needed.

These were all custom designed. Until I got to CommSci, I didn't deal with the concept of a packaged program at all.

Abbate: So you would start with the output

Johnson: Yes, in most cases.

Abbate: And prepare a flow chart?

Johnson: Yes. Usually you'd draw a rough flow chart of the information flow. You're supposed to do flow charts of every program, too, but that was ...

Abbate: More theory than practice?

Johnson: Yes. It's a good theory, actually. It's something that IBM really established as a standard. The problem is in so many cases people are working under time pressure and it was really faster just to sit down and write the program. And very rarely did I ever write a program from scratch. If I had to produce a report, I'd find another program that had a similar report and take that deck of punch cards and take it over to the duplicator. I'd make an exact copy of that program and then I'd start pulling out cards here and there and replacing them. I usually used colored card stock so I could tell which ones I had replaced. Rather than writing everything out on a coding pad, it was much faster to just take an existing program and alter the parts that need to be altered.

Okay, maybe there was a flow chart for this program, but why am I going to take that and erase it and change it? That's how everybody worked. Maybe IBM programmers sat down and drew out the flow chart on all the steps and then wrote the code. But mostly when you had to get the program done, the fastest thing was to take one that was similar and go from there.

Abbate: What were the pluses and minuses of working with COBOL?

Johnson: I can't compare it with too much in the sense that I never really used the other languages. Certainly one of the pluses was the ability to do just what I said -- that it was a standard and there were lots of ways to steal pieces from other things and patch it all together

The justification that was always given for COBOL was that it's like English. But that's bullshit. It's as much a programming language as any other. You just have to write more characters to code an instruction. But I got so used to it. That's what everybody expected, and unless you were doing something very strange and esoteric, nobody ever wrote anything else. So I just worked with it.

One of the minuses was that people didn't work within standards. This was especially true at the County of Alameda where they really didn't have any programming standards. So one guy would write "Add A to B equals C" to make COBOL into a formula-like language. Somebody else would write out very long descriptive data names using as many as 30 characters which made maintaining it very cumbersome. I maintained one set of programs I wish I had saved copies of because nobody ever believes me. The programmer gave the data fields names that would rhyme when he wrote the instructions in the Procedure Division. Honest to God! He obviously had too much time on his hands! He himself couldn't figure out a few months later what the fields represented because the data names weren't descriptive. It was really stupid. But that's not really a flaw of COBOL; it's a flaw of not having the standards for how to write COBOL, implement it or enforce it in the shop itself. As an entry-level programmer who had to maintain programs written by other people without any consistent standards, I learned a lot about the importance of making it easy for a maintenance programmer to understand a program very quickly.

STARTING ARGONAUT INFORMATION SYSTEMS

Abbate: You went into business for yourself in 1971?

Johnson: Yes. There's a story there, too. The company I was working for, CommSci Systems, was going out of business. This was the first place where I encountered the concept of packaged software and I really believed in the concept. The man who had founded CommSci and was running it was very flaky. A lot of good programmers are very flaky. He was a good programmer and he was very flaky. He had all kinds of financial problems and one thing and another - a very complex story. He also had some ex-wives who were after him for money. At one point he decided he couldn't deal with it anymore; he was going to leave the country.

So he came in one day and told his four employees that he was shutting the operation down and that we'd all have to find new jobs. I loved the job and believed that there was a big future in packaged software. It would have been easy for me to get another job because of my experience. But I thought I'd really like to go on doing what I was doing.

So I decided that what I would do was form another company and negotiate the right to go on selling the products that CommSci had, which were the payroll system and a beta version of the accounts payable system. But the person I had to negotiate with was leaving the country so I had to do all this quickly.

One of the lawyers I had worked for had remained a good friend and he was very helpful. We were trying to get the new company incorporated, and in California, as I guess in a lot of states, you have to file the name with the Secretary of State, to make sure it's a unique name in the corporate index. There's also a policy that if you think you're going to form a company, you can have a name on reserve for three years or so. So I kept coming up with all these hopelessly trite names like Business Data Systems and everything I came up with was already reserved. And we were running out of time.

The office where I was working in San Francisco was right on the edge of the Tenderloin district. I parked my car in one of the lots in that low-rent district, because it was cheap. So we were down to the wire, and at lunch time I went to get something out of my car and walking back to the office, I walked past the Argonaut Pawn Shop. True story. Argonaut. I walked in and called the lawyer's secretary and asked her to see if she if she could reserve Argonaut Information Systems. It's a good name, starts with an A; it's at the front of the yellow pages.

She called back and said "You have it!" So within 24 hours we had the paperwork done. I actually went down and stood on the street with the papers for the guy to sign. He had his car loaded up to take off for Canada or Mexico or whatever. He pulled over, signed the papers and that was that! So the company was named after a pawn shop on 5th and Mission in San Francisco!

Abbate: And I thought it was your literary background!

Johnson: No - nothing to do with that! And it worked out fine. It gave us a nice little logo and the A listing helped. And Argonaut also has a lot of meaning in California in terms of the gold rush pioneers who were Argonauts.

Abbate: So your first products were inherited.

Johnson: Right. The payroll system had a number of customers, but the accounts payable system had just one customer because it was in the process of being developed in partnership with that customer. Of course, that customer was very anxious to keep this going, too, because they were well into the implementation stage, so they were very supportive. The data processing manager there was a big help to me, too. So I ended up with those two products that became the products of Argonaut.

Abbate: This was called packaged software at the time, and it seems to be sort of intermediary between totally custom software and off-the-shelf. It sounds like you were developing it for a particular client and then built on that

Johnson: Yes, actually that was very commonly done, to finance the development of a product. You would work out a deal with a customer that needed the product and develop it, and they in essence would pay for the development, but then usually what you would do is reimburse them for part of that cost from future sales. A lot of people did that.

The product Mark IV from Informatics, which was one of the first big software products, was financed by five customers. You couldn't get money from outside sources to pay for the development of a software product, because nobody knew what a software product was. So partnering with the first customer was a common approach.

The payroll system had actually been financed before my time by a customer, on the same kind of a deal. The man who founded CommSci Systems and the programmer who worked for him had gotten a customer to pay for development of the payroll, but then they went out and resold it to other customers. But it wasn't totally an off-the-shelf sale, because in almost all cases customers required some level of customization. Most people who were in the packaged software business in those days, in the applications area – this was not true for systems software -- would usually end up modifying the software to some extent, working from that basic package.

In my opinion, the man who was running CommSci was doing too much modification for the customers. Your margins on the services aren’t nearly as good as they are on the product. One of the things that I wanted to do when I took over was to figure out how to make the product more generalized and also how to sell it differently so that people weren't expecting that they would have to modify it as much.

So I wanted to really focus on making it much closer to the product end of the range than to the services end.

Abbate: But what your business sold to the clients was a kind of package? They also got installation and a certain amount of training?

Johnson: Yes. They got the source code, which again was pretty standard for applications products. They got user documentation and they'd get some tools that they needed to install it.

Pricing was all over the place for similar products from different companies. But the way I worked my pricing was that the quoted price included a couple of days of installation, but it didn't include any modifications. So if people wanted modifications, then that would become part of the original contract, but that was always an add-on amount. It was a fixed amount usually. I tried to estimate exactly what it was going to take to do it. But it usually worked out pretty well. Partly because, if I ran into a customer that needed lots and lots of modification, I just backed away. There were plenty of other companies that could do that. I really didn't want to do that kind of thing, because I really wanted to stick with selling it pretty much "as is."

It also makes it a lot easier to support it, if what the customer has is very close to the original. And the customers would always change the code themselves. I mean, they've got the source code and they would change it as time went by. Their users would want a different input format, or a different report format. It became very hard for you, as the vendor, to support it, because you didn't have any record any more of what they'd done. So the more I could keep it standard, the less I had to deal with support issues.

SUPPORTING THE PRODUCTS

Abbate: What kind of support was included?

Johnson: With a payroll system you have the problem of tax changes. You have to provide updates for that. So for the first year I included any tax changes that came through. It's usually once a year, but sometimes one of the states would make an extra tax change and you would have to do it in response to that situation.

So they got any tax changes that came through the first year and for a year they could have telephone support. They could call up and we would help them solve problems over the phone. If they wanted somebody to come on-site, that would cost extra. Then after the first year, they could buy the tax update service only. Everybody did because there was no point in their trying to keep track of the tax changes when we were doing it for 100 other customers

Most of them also bought ongoing telephone support after the first year which meant that they could continue to call and we would answer their questions.

Abbate: So if they wanted upgrades, they would pay for them?

Johnson: Yes. There were required statutory upgrades such as changes to the W-2 forms which required changes to the print program. If you paid for support every year, you'd get any changes that were along those lines.

And we did send bug fixes. If we found that there was an error, we would send out changes to the source code. Then in some cases, as with every software company, a customer would request a new feature. If it was something that we felt was an enhancement to the system, we would often make the change and use that customer to beta test it, then we’d make it part of an upgrade that went to everybody. My goal was always to try to keep the source code as standardized among all the users as possible so that there was a chance of figuring out what went wrong when they called with a problem.

Abbate: It seems like one of the secrets of your success was that you made the code self-maintaining in a way.

Johnson: Yes, we tried to.

Abbate: Where did you get that or was that a departure?

Johnson: Well, yes and no. Of the two guys that originally wrote the payroll system, one of them was one of the most brilliant programmers I've ever met. He just naturally knew how to write elegant code. They were both good programmers, but one of them was able to write things in such a way that it was clear when you looked at it what the code was doing. One of the things that made maintaining the tax routines so easy is that he designed it in a table format, so there was virtually nothing in the code that had to be changed. In almost all cases, whenever there was a change to a tax calculation, it was possible to change something in a table, very easily done. A very bright guy.

One of the people who started working for me in the 1970s, supporting that payroll system went to Ross Systems when Argonaut's products were sold to Ross much later. Then the payroll system got sold to yet another company, and she moved to that company. In other words, she continued to work with that payroll system through its ownership by several companies. I mentioned to her recently about how much I had learned from the way the original tax routines were written and she said that there were still companies using exactly that same code that was written back in 1967, 1968, 1969.

It was just such good, strong, elegant code. Anyway, I learned a lot from him. I looked at the things that he did and I saw why it worked. So I ended up using a lot of his style and incorporating it throughout the programs. That was really my model - the part of it that he had written. I just adapted that to everything else.

Abbate: So then your users could also figure it out.

Johnson: Yes. In addition to adapting to his style of coding, I also made very, very heavy use of the copy statement in COBOL. I set up lots of standard routines that were incorporated in all the 20-some programs in the payroll system by use of the COPY statement which allows the compiler to pull that code into the program from a library of routines. I used it not only for I-O routines and the Data Division which is commonly done, but also for Procedure Division routines. Which was pretty easy to do in these accounting systems since the majority of the programs were just variations on ways of producing reports.

I tried to get to the point where at the most twenty per cent of the program was unique to that program, but everything else was coming in from these copy libraries so it was the same in every program. So when a maintenance programmer looked at the code to debug a problem, he could just focus on the part that was unique to that program. Everything else was working in other programs so could be eliminated as the source of the problem.

This technique does not produce the most efficient programs. These programs didn’t run as fast as programs that were customized for each particular function, but a payroll system is all tied up in reports anyway.

So it didn't really matter that you were losing some efficiency in run time. What you gained was efficiency in the programmer time to try to figure out what was going on when problems occurred.

ARGONAUT’S CUSTOMERS

Abbate: Who were your clients? Big companies? Small?

Johnson: Medium-sized companies. Initially that came about because we were working with the IBM DOS operating system instead of the OS operating system. The DOS system ran on the smaller models of the 360 which by definition were for smaller companies. But then I made a decision to stay with the mid-sized company market. In the mid-1970s, most software companies considered the Fortune 500 the marketplace to be in. That's where the big companies were; that's where the big budgets were for data processing.

So all the companies who were selling a payroll system were competing against each other in that marketplace. So I thought the next tier of companies below that was worth going after. And there was much less competition there. What competition there was came from local programming services companies that focused on customers in their geographic region. But in terms of competitors who had a real product and were competing nationwide, there really wasn't much competition in that space until later in the 1970s when there was a large enough market for software for minicomputers.

So I went after mid-sized customers – those that were big enough so they had a computer and where they had enough employees to need a payroll system, but below that top tier where everybody else was beating each other up trying to compete.

Abbate: They were still using mainframes - the smaller ones?

Johnson: Yes, like the System 360 series – the lower end computers were the 360/30 and the 360/40. As I recall, it was the 360/30 and the 360/40 that used the DOS operating system, then when you got to the 360-50, it used the OS operating system, which required a different set of job control language. The COBOL would still compile on either, but the commands that you used to set up the jobs would be different.

We had some customers that we ended up with on the upper end too, that had the OS operating systems. But most of our customers were companies with the smaller computers. And that worked out well - it was a good market to be in.

Abbate: Was it separated by platform or by price or features?

Johnson: Mostly by platform. In the early 1970s, the only market that really amounted to much was the IBM marketplace. It was the only place where there were enough computers out there to make a big enough market for a generalized package like mine. It was different for software companies that were in vertical markets because different hardware companies specialized in different vertical markets. If you were selling a point-of-sale system, then you were very likely in the NCR market because they were dominant there. There were lots of NCR computers in that kind of retail environment.

But for my company, it was really the IBM market, at least in the beginning of the 1970s. That's where there were enough companies with computers that needed payroll systems. If you figure at any point in time, 1 out of 100 companies needs a new payroll system, you need a base market of several thousand companies in order to sell even a couple of dozen of payroll systems a year. And in the early 1970s, the IBM market was the only one that was large enough to support a company like mine.

In terms of our pricing, we priced the same regardless of the size of the computer. Not everybody did that. Most software companies followed IBM’s lead and charged more for the same software if it ran on a larger computer. But as far as I was concerned, our costs to support the product on a larger computer was the same as on a smaller computer, so there was no reason for the price to be any different.

Payroll was the first back-office application where there was a big market for a software package because it was more standardized than other accounting applications. There were external factors such as the taxing authorities that determined how payroll processing needed to be done, so it was a door-opening product for many software companies since the customers found it easier to accept buying a standardized payroll package than other kinds of applications.

HIRING EMPLOYEES

Abbate: Tell me about your early employees.

Johnson: At the beginning, I primarily needed people to go to the customer’s site and install the payroll system for them. I didn’t initially hire anybody to do development work because I was rewriting all the programs myself since I had such specific ideas about how I wanted it done. But almost all customers wanted someone to come on-site to install the system and run some tests which usually took about a week and almost always required traveling.

I did a lot of those installations myself, but I had a family and didn’t want to be traveling all the time, so I found a couple of contract programmers who were very good at that kind of work and liked short-term projects. They did a lot of the installation work for me for the first several years.

I had an office in downtown Berkeley and the first employee I hired was my sister who moved to California to work for me as a secretary/office manager/etc. She was stuck alone in the office a lot of the time because I was out installing systems or making sales calls.

One day a college kid named Paul Gustafson walked into the office. He had taught himself to program when he was going to City College of San Francisco and had transferred to UC Berkeley for his junior year. He was looking for a programming job and of course Argonaut was the first company listed in the yellow pages. So he walked in the door and ended up talking to my sister for over an hour. I think she was just glad to have some company since she was alone in the office so much of the time but he really impressed her and when I got back from my trip she insisted that I had to hire him and would really regret it if I didn’t.

Using contract employees was working well for me, and I hadn’t planned to hire anyone for awhile. But I called him up and interviewed him. He was looking for a part-time job and convinced me that I ought to at least try and see if I could turn some of the development work I was doing myself over to him. He turned out to be great, and after he graduated from college he came to work full time and ended up being in charge of all development even though he was very young. By that time, I had hired several other people to help with customer support and then the company just grew from there. Paul went on to have a great career in the industry as a software development manager in some very large companies.

USER REQUIREMENTS

Abbate: You said one of the most challenging things about developing the software was just figuring out what the user's requirements actually were. Was it sort of a one-time thing for each product or client?

Johnson: That was the case in the years before I started working with software products. In my previous jobs, the systems were custom designed like the bill-of-lading system for the freight forwarding company. Previously, all the records had been kept on ledgers that were stored in a bin. So it was a matter of trying to get them to think through what they did. When things aren't automated, people tend not to handle each transaction in exactly the same way, and often for very good reasons because things don't tend to always come to your desk for handling in the exactly same way.

So in that environment you had to sit down and get people to think through what was common about what they did, and if there was a variation in the way something was processed, there had to be for a reason. It's very difficult to get people to think that way, so that was hard work.

With the software products, you go in with the procedures already defined and it’s a matter of whether or not the procedures you are offering in your software are a close enough match to the way that potential customer does things. With a software product, the customers almost always have to make some changes in their procedures and they don’t want to have to make very many. I was very fortunate in that the payroll system had really been well designed and it was a matter of trying to understand how close the match was between what our product offered and their requirements because we weren't going to make a lot of changes if their needs were different.

But before I started working with software products, developing a system wasn't really a technical challenge because it was pretty easy to write the COBOL once you knew what it was supposed to do. The hard part was getting people to really define accurately what it was supposed to do because they aren't used to thinking that way. People don't normally think systematically or think about why they do things the way they do them. It was fun, but it wasn't a technical challenge so much as getting people to rethink the way they worked. That was the challenge.

Abbate: It seems like customers had to change their procedures every time they bought a software product.

Johnson: That's true. And that was a challenge for everybody who sold application products. The users resisted bringing in a software package because they did have to change the way they worked. Most of the brunt of that resistance would be borne by the DP manager because that was the person who made the decision to buy a product instead of developing a custom system so that they could have exactly what they wanted.

And the DP manager always had a very strong economic argument about how much cheaper it was to buy a product. I remember one case where we sold an accounts payable system to a company and one woman in the accounts payable department just refused to believe that any of this was going to work. It had to do with the way she coded the invoices to identify which expense accounts it should be charged to. We provided a form they could use to enter the expense accounts and attach it to the invoice. But this woman just insisted she could not use that form. So I watched her work for a while, and she used little green slips of paper to note the expense accounts and she would staple those to the invoices. So I reduced the size of our form and printed it on green paper and then she was just fine with it. When I gave her a form that looked like what she was used to, it was okay.

Abbate: I think you mentioned people were worried that they might lose their jobs. It seems like the economic argument would be that it's cheaper to buy a product because you can get rid of a lot of people

Johnson: Well, it was cheaper than hiring extra programmers.

But it was the clerical people that always were worried about losing their jobs. But it wasn’t uncommon for companies to end up hiring more clerical people, because suddenly they had more information to work with. They would start getting more granular in terms of the kind of data they could keeping track of. Now all of a sudden not only can we keep track of expenses by account, but we can also start assigning job numbers so we can see what our expenses are by job. But that created more work for the people who were coding the invoices so in many cases, it actually led to additional people getting hired.

The primary economic argument for the DP manager was that he didn’t have to hire more programmers to develop a system from scratch. He still needed programmers to maintain the systems after they were installed but it saved a lot of money in development costs. So he had a great economic argument, that he could usually defend, for buying a product.

Abbate: Everyone was happy.

Johnson: Everyone was happy. Very happy.

LICENSING OTHER VENDORS

Abbate: At some point you licensed your product to third parties?

Johnson: Yes. It seemed like a good strategy to me because I felt I didn't know marketing. But I didn't come up with this idea on my own. There was a company called InSci, short for Information Science, and they sold a fairly complex payroll system that was much higher-priced than mine. At one of the ADAPSO conferences, the president of the company came up to me and he said he’d like to make a deal with me. They didn’t want to have to keep up with all the tax changes so he proposed imbedding our tax routines into their payroll system and paying us a royalty every time they sold a payroll system. And then we could have the maintenance revenues from the tax update service after the first year.

The way we handled keeping up with tax changes was to subscribe to a publication from Commerce Clearing House that reported any statutory tax changes for every taxing jurisdiction in the U.S. So we'd get those reports and know what changes we had to make to the tax routines and since we were doing that in any case, the agreement with InSci wouldn’t require any extra work on our part. So it sounded like a really good deal to me because they paid us a lot more for the tax routines embedded in their payroll system than we could have gotten for the tax routines on our own. And then we had this additional maintenance stream coming in just for the cost of mailing the updates to more people. So that worked well.

So I became open to that idea, and then a few other people approached me. By about 1976 or so, we began to see a software market opening up in minicomputers. We had an edge in the minicomputer market because of the simplification I had done to the COBOL programs in order to make it easier for the maintenance programmers. That also meant that it was easy to port it to computers that had a different COBOL compiler.

So I began to go after re-licensing opportunities pretty aggressively and we ended up with re-license agreements with a number of different software vendors. There was a point in the late 1970s when there were five companies in the U.S. that were selling payroll systems to run on the HP 3000. It was all the same payroll system, because the other four were licensees of the Argonaut payroll system.

So if you took the source code from those five different payroll systems and looked at the listings side by side, you’d find out that the source code was identical. One of the companies selling it was Hewlett Packard itself. ASK, the manufacturing software company, was selling it. A company called Software International out of Boston was selling it and a company down in Florida called Collier Jackson. If you looked in the ICP directory for payroll systems that ran on an HP 3000, those four companies would be listed along with Argonaut. But it was all exactly the same source code.

But there were legitimate differences between them. They each packaged it differently. HP included a lot of support with theirs. It was very high priced compared to ours, but it included all the HP support and had very slickly presented user manuals. ASK interfaced the payroll with their manufacturing management control system called ManMan. They were keeping track of labor costs, but then they embedded the payroll system so that information automatically became input to the payroll. So it was an add-on to that.

Software International was selling a general ledger system, and they interfaced the two so that the payroll system would automatically feed into the general ledger. And Collier Jackson, also did something to make it unique, though I don't remember exactly how they differentiated their version.

I think it really shines a spotlight on what a software product is. Because it's not just the source code. The source code was identical but each of these was a unique product going to a particular market. They put their own brand on the documentation. So a software product is a lot more than just the code. We got a royalty every time one of those got sold. It was pretty nice.

Abbate: Do you have a rough idea what percentage of your business eventually came from licensing?

Johnson: I think probably it got to the point where it was about 25 or 30% of the revenues. It was a much smaller number per unit than we got from direct sales, but of course there were marketing costs involved with direct sales that there weren't in the licensing agreements. It differed from company to company. Hewlett Packard also provided all the support to their customers, so when they sent us a check, it was a full royalty check, period. With other companies like Collier Jackson, their customers had the option of contracting with us for maintenance support and other services. So they weren't all identical. The problem with Hewlett Packard is that they didn't sell very many. They weren't very good at selling application software. But when they did sell one, it was clean.

Abbate: Did most of the third party customers also subscribe for updates?

Johnson: Well, the Hewlett Packard customers subscribed to the service from HP and, as I recall, we charged HP a higher price to provide the service to them so they could provide it to their customers. But as far as we were concerned, what we sent them was identical to what we sent anybody else. It was just another label we printed when the package went out.

PORTING TO OTHER COMPUTERS

Abbate: So when did the minicomputers start coming in -- was it late 1970s?

Johnson: Yes, in the late 1970s we really began to see them out there in the commercial marketplace with customers looking for accounting packages that would run on them. You started to see DEC systems and HP was very successful with the HP 3000. It was a good, mid-range size, mid-priced computer.

One of the side effects of having simplified the code so much was that it was easy to modify for different computers. Many of the COBOL compilers on the minicomputers would only process a subset of the full COBOL that you would get from IBM, but that was fine because I had stripped out everything but the most simple, straightforward COBOL instructions to make it easy for the programmers to understand.

There were always some differences in the COBOL that ran on each computer, like the descriptions of the input/output devices which were always unique on every operating system. But because we had so much of the code contained in the copy library, all we really had to change was the copy library. Paul became very good at writing programs that would read our standard source code and modify it to compile on a different computer. Maintaining our source code wasn’t very exciting because my standards for how the code could be written were so strict. So this gave him a chance to do some really clever programming to write those filter programs and he never found a compiler that he couldn’t automatically do the conversion for. So whenever we had to deliver a payroll system for a computer other than the one we were using to maintain the master copy, we would just run the code through a filter program and send it out. That meant that we didn’t have to maintain a dozen different copies of the master code on all those different computers. We could do upgrades on the master copy and then just run it through the filter programs to send it to customers in the variation that they needed.

At one point we had 16 filter programs that could convert the code to run on 16 different platforms: Perkin Elmer, Prime, General Automation -- computers you've never even heard of, right? But we could deliver. We were set up to do that and it worked fine.

COMPETITION AND MARKETING

Abbate: How was the competition on that?

Johnson: There were now more companies that were aimed at that size company -- where the minicomputer was. There were thousands of software companies, as opposed to hundreds.

When we were going into a sales situation, we were much more likely to find that the customer had several options. At the beginning, the only real competition we had was the customer deciding they'd rather do it themselves but by the end of the 1970s, that changed. They had several choices in terms of their products

Most of the competition was companies specializing in DEC computers and they usually were not as productized as we were. They had some payroll programs that would run on DEC and they would come in and deliver a much more customized product. There were fewer that were delivering it straight off the shelf. But it was tough competing because they were offering a much more customized, tailored version. So we did lose sales in that case, when somebody came in and said, “I'm a local company; I'm right down the street. DEC is all I do; these guys do everything else. I really know DEC and I'm going to be able to customize for you. “

But it didn't bother me too much because I never did want to do those kinds of contracts anyway. But that was one of the areas of contention with the employees as I began to get staff. They felt that I was missing out on opportunities; that there was money to be made there. And I think they were right, but I just didn't want to. I was content with what we had, you know?

Abbate: I guess at that point you had a marketing person?

Johnson: Yes, and actually he did some of the biggest re-licensing deals -- the Hewlett Packard deal for instance and the deal with Univac. He’s a very bright man -- one of these people who is very articulate and always has an answer to every objection, able to argue through anything.

One of the re-licensing deals he did was with Univac. I'll never forget the meeting because Univac brought in five attorneys to work out this contract. And on our side was just Bill Hixson, not even a single attorney representing us. The five attorneys just didn't have a chance against him. For everything they tried to put in the contract, he had a perfectly good reason why it should be this way or that way. He was just very good at that.

He contributed a lot. He also created a lot of tension in the company because he wasn't interested in just off-the-shelf sales which he felt were too small. He wanted to go for bigger deals and bigger contracts and that really resonated with some of the employees who felt that I was holding them back.

It got to a point in the late 1970s, where there was a lot of pressure on me to change the way the company was operating. My second marriage was coming apart which had a lot to do with the fact that I was so absorbed in my work. My husband had a lot of complaints about that – legitimate complaints -- but in the long run it turned out that the problems went much deeper.

So I was trying to deal with that and the fact that there was a lot of restlessness among my employees. At one point I made a list of all the things that I saw had to happen for the company to grow enough to satisfy the career ambitions of the people who worked there. I also made a list of the things I wanted to do and enjoyed doing, and there wasn't much match between the two. I realized that I either had to shift my priorities to do things differently, or I had to get out of the way and let the rest of the people do what needs to be done. That’s what I decided to do. Bill Hixson put together a deal so they could buy me out. It didn't happen instantly; it was over a period of several years actually. But at that point he really took over much more of the strategy and the marketing.

Everybody loved it because there was more money coming in and there were bigger contracts. Everybody was working harder, too. There was more work needed to do this, but it was fine for everybody and it allowed me to ease my way out. It worked out okay even though there were some tough times and tension.

But it really began when I brought someone in to take over marketing because he saw opportunities that I either didn't see or didn't want to pursue. He knew how to bring the rest of the people along with him to pursue those.

Eventually it led to my selling the company, and then that core group of employees did very well later on when the products were sold to Ross Systems.

DEV ELOPING A GENERAL LEDGER PRODUCT

Abbate: Were you developing new products while you were still in the company?

Johnson: The one new product that we developed was a general ledger system. It seemed like a logical add-on to the payroll and the accounts payable.

Abbate: How did that fit with your other products?

Johnson: Well, it was a simple general ledger system which kept track of all revenues and expenses and produced the balance sheet and the profit and loss statement needed to run a business. For most companies, most of the expenses are being tracked by the payroll system and the accounts payable system, so it makes sense to have a general ledger system that is compatible.

Not wanting to start from scratch completely, I acquired at very low cost a general ledger system from a Canadian company. They weren't going to sell it in the U.S. And I ended up again rewriting it to fit my standards, but what I got from that was that the functionality was well defined. I don't think there was a line of code that was left once we got done rewriting it, but I didn't have to try to figure out what it should do. That was already all defined.

I released that product in the late 1970s. The problem was that there were already people well established in that field like Software International, based in Andover, MA. They had a lock on that market. They started out with general ledger as their first product so they had a lot of satisfied customers.

We had some sales. Most of these were to people who already had one of our products and our general ledger system automatically interfaced with our other products. The payroll system and the accounts payable system automatically generated all the entries for the general ledger system, so you didn't need to do any customization. So we had good add-on sales from that, but not from getting new customers just for the general ledger system. Maybe one or two - not very many. So it never did as well as the others because it was a follow-on.

Then at one point we did an inventory system, but it wasn't a true inventory system in the sense of keeping track of items in inventory. It was the piece of the inventory system that goes into an accounting system, but wasn't a full inventory system in the sense of keeping track of items in stock and notifying you when items are running out. What we had was just a way to record inventory data for the general ledger system and, again, it was an add-on that people bought. But we didn't sell it to new customers.

Abbate: So did you learn anything from the less successful products?

Johnson: Yes, I guess what I learned was that it's very difficult to get into and really make a success with a product if you're not one of the first several companies to get out there. It was very difficult to plow new ground. Selling an add-on to existing customers was relatively easy to do. But to go up against established companies that had been out there for several years with a general ledger system that had a long list of satisfied users was very tough. You had to be there first, or have something really different and unique.

MANAGEMENT ISSUES

Abbate: What kind of management issues did you have?

Johnson: I always had the management issue of trying to get people to stay with what I felt were the necessary standards. I had this very definite vision of how a software product should be built. But that required really putting a limit on how the programming would be done. It was very formulaic. If we were going to add a new feature to the system and add a new program to do that, it had to be written in exactly the same style. That was always an issue to get people to want to or be willing to do that and understand the need to do that.

There were issues related to supporting customers because that was an important part of what we provided, the telephone support. But the people who did the best at that work seemed to understand how important it was to give good customer service. I don't really recall having a lot of problems with that. I think that's just that you set an expectation that if somebody calls for help, you give them help. I don't recall anybody really having problems with the standards we set for that.

I do recall once when Bill decided we needed to build some new systems for more specialized applications that we tried to hire some entry level programmers. This was really my idea -- to get people before they had established their own style of programming. But some of them just turned out not to be trainable, and we had to give up on them. That was hard.

For the most part, it was a matter of mutually recognizing that this isn't working. I can't think of many situations where it was just a matter of: You're out of here.

ADAPSO

Abbate: You joined ADAPSO fairly early on, right? Were you a founding member?

Johnson: No, I joined ADAPSO in about 1973. The person who sold me on ADAPSO was Larry Welke. He got to know all the software people because he was selling them on listing in his catalog. He'd come around and visit once or twice a year: "I'm in town; let's go have a drink." He told me that I really should join ADAPSO because I'd learn a lot from those people and I kept saying, sure, sure.

ADAPSO had a conference in San Diego – I think it was 1974 -- which wasn't that far to go. So I went to see what it was all about and discovered that he was right – I really could learn a lot. I walked into a room where they were discussing the California sales tax issue. And this was a big deal because if you delivered software electronically over the telephone line, there was no sales tax. But only big companies had the capacity to do that. If you delivered it on punched cards or tape, which is the way we usually delivered, then there was sales tax on it. It was really penalizing the small companies. The tax authorities just didn't understand that what people were getting delivered over the phone lines was exactly the same thing that we delivered on magnetic tape. But because it wasn't on a tangible medium, there was no tax due.

So here were other people that were California software companies and they were organizing to campaign the State Board of Equalization to make them understand that the sales tax laws were discriminating against small software companies. And I knew I wanted to be involved. I knew that the tax laws were unfair but I didn't know what I could do about it. And here was an organization that was going to try to do something.

Larry almost always did a session at the conferences on selling and marketing -- that was his specialty. So right from the beginning I realized there really was a lot I could learn. One of the things that was so great about the organization in those days is that people were so willing to share what they had learned. You'd sit in a panel discussion and someone like Tom Nies, the founder of Cincom, would talk about a mistake he'd made and share what he'd learned with people who were potential competitors. There was just this great sense of being in this together trying to figure out how to make it work in an industry where there weren't any tried and true business models.

So I got hooked on it very early because I realized that I was really learning a lot and it was fun. They were fun people.

Abbate: Were there a lot of women in ADAPSO?

Johnson: Very few. I got to know a lot of journalists. There were a lot of women journalists and we'd always end up in the restroom together. Everybody always wondered why I got so much press coverage. Because I got to know them! How come they're always quoting Luanne? Well, because Angie Pantages and I were talking in the bathroom.

There were a few. There was a woman, Ruth Coolidge, who was running a service bureau company. When I first started with ADAPSO, the first thing we would do the first morning of the meeting was that everybody would stand up and introduce themselves and tell what they did. So you're talking about maybe 70 or 75 people. It wasn't long after that that they had to quit that because it started getting too big - several hundred people came to the meetings. But for the first couple years, it was an intimate group. And I suppose out of those 70 or 75 people there were 2 or 3 or 4 women - something like that. Not a lot. And all the others were running services businesses.

I didn't find any other women that were running software products companies. Sandy Kurtzig, the founder of ASK, was selling software products but she wasn't interested in coming to ADAPSO. Connie Galley was working for a software products company, but she wasn't a founder, and she didn't start showing up until later.

I'd look around and see another woman taking notes and I'd assume she was a journalist. A lot of the technology journalists were women: Angie Pantages, Becky Barna, Sheila Cunningham. There were quite a few women from the various technology journals who would cover these meetings. They were good writers and good technology writers.

Abbate: Why?

Johnson: I have no idea. I would love to try to figure that out sometime.

Abbate: And you were with ADAPSO for a long time?

Johnson: Yes, I was on the Board of Directors of ADAPSO for a while. Then we went through the whole transition with Argonaut and Bill was gradually taking over more and more. I ended up in a consulting relationship before I was completely out of Argonaut. I had a retainer agreement with them at the end.

But I was doing other consulting stuff, too -- working with other companies and doing odds and ends. Anything that came up -- everything appealed to me. The man who'd been running ADAPSO for 19 years, Jerry Dreyer, resigned very unexpectedly in 1986. A couple of the people who were on the board were people I had worked with in consulting roles. They had to find somebody to come in on an interim basis.

So they called me up and asked if I could come to Washington for a couple of months to keep things in place until they found a replacement for Jerry. They said "Can you come to Washington for a couple months?

It happened that I had just finished a contract and except for the ongoing retainer agreement with Argonaut which was taking very little of my time, I didn't have any other big commitment. So I agreed, but said that it would have to be two months at the most. They said "Okay, no problem." I said "When do you want me there?" They said "Well, tomorrow."

So in a couple weeks I went back there. The two months dragged on because it took them longer than that to find a replacement, so I was there about six months. They finally found a replacement; and hired him.

During that six month period, in the Fall of 1986, we celebrated the 25th anniversary of ADAPSO. Part of the celebration was forming the ADAPSO Foundation to fund projects to use computer technology to help people with disabilities. It showed we were a mature enough industry to give something back to society.

I was in the office packing up to return to California when the Board of Directors of this new foundation was meeting in another room. They called me in and asked me to run the foundation on a part-time basis so I could continue my other consulting.

I tried to negotiate running it from California but they argued that since the ADAPSO staff would be supporting me, it wouldn't work unless the foundation office was located with the ADAPSO offices in Arlington, VA. I thought about it for a couple of days and then said I'd do it for one year, just to get it off the ground. I felt kind of responsible for getting it going since I had been involved in all the planning to get it set up.

I'd never thought about working for a non-profit organization, but I just loved it! It was really exciting. It's great to see somebody get their payroll system running but when you see a kid that can't talk suddenly communicating with a simple little computer device – now, that's exciting! So it was really a lot of fun to see applications of this technology go to people who saw some immediate benefit from it and see that it really changed their life!

So I was really having a good time with that and the one year extended into two years. Then, they decided they weren't happy with the man they had hired to replace Jerry Dreyer, so they came back and wanted me to take the Executive Director position permanently. That was in the spring of 1989. At that point I'd been living in Washington for about two and a half years.

I wasn't sure if I really wanted to do it, but I knew that if I didn't take the offer, I'd always wonder if I should have. So I said I'd do it. That was 1989, and it lasted till 1995. That was the period during which it changed from ADAPSO to ITAA and it was a very transitional period for the organization. But that's how I ended up there -- there was no plan to it, it was just one thing led to the other and I never had the courage to say no because I was always afraid I would make a mistake by saying no. If I made a mistake by saying yes, I can fix that! But you can't fix a mistake by saying no, because then the option is gone. So it was just a matter of -- I might as well see what happens.

And I was glad. It was very exciting and a great opportunity. I got to meet all the people who are still an important part of my life and that I'm using those contacts to do what I want to do now. Half the people that are here at this meeting are people that I know through ADAPSO.

Abbate: So by the late 1980s, were the members of ADAPSO mainly PC oriented or was it mainframes?

Johnson: No, it was still mainframes. ADAPSO never successfully recruited the PC people. At one point in time they tried to and added a special division within ADAPSO to address the issues that the PC people had. But their most significant issue was piracy. They had a problem with international piracy, but also piracy within US companies. Companies would buy one copy of the software and install it on multiple PC computers. It wasn't a deliberate decision to pirate the software. But they'd buy a copy and it would proliferate around the office and they wouldn't be paying for it.

So the companies that were trying to address this were Ashton-Tate and MicroPro and other successful early PC software companies. I'm not sure if Microsoft was involved at that time. They decided that the only way to stop this was to have some very visible lawsuits. Through one means or another -- this was before I was president so I don't know for sure what was going on then -- they were able to establish a clear cut case. It may have been information from a disgruntled employee or some other source but they knew they could prove that there were copies being used that had not been paid for within that company.

So they went to the ADAPSO board and they said they wanted to sue this company because they knew they could prove that this was happening. Well, ADAPSO had grown initially from small companies like service bureaus and software companies and the professional services companies and by this time there were some large companies like GE and IBM who were in the network business and were ADAPSO members. And the particular company they wanted to sue was a major customer for several of these large ADAPSO members. IBM was one of them. They said, "We're not going to sue our customer; let's negotiate with them; let's explain to them what the problem is." And so on.

And the PC software guys said, "The only thing that's going to work is if we sue them!"

At about this time, Ken Wasch had started an association called the Software Publishers Association that was focused on the PC software companies and he had no problem with the idea of suing the companies where illegal use of copies could be proved. So SPA took on a very aggressive piracy program. They'd go in with a midnight raid with the sheriff and determine how many copies of the software were on all the computers and then demand to see the purchase orders for that many copies. Usually they would negotiate a settlement. But they were willing to be that aggressive and that really mattered to the PC software companies. They didn't have the same kind of long-term existing relationships with big companies the way some of the ADAPSO members did. Of course ADAPSO was hamstrung. As a result of that, SPA became the organization that represented the PC companies instead of ADAPSO.

People who were cutthroat competitors would come to ADAPSO meetings, but they always treated each other with respect there. There never was anything personal about it. It was a really good lesson in how to be a good competitor and respect your competitor. Be good friends with them and share information and then go out the next day and try to kill them in the marketplace.

It's not like that anymore. I mean, the industry has changed so much. But in the early years it was an incredible, incredible environment.

CONSULTING

Abbate: So you did some consulting, even maybe during your time at Argonaut and certainly after. What kind of companies did you consult with?

Johnson: Mostly similar companies - software products companies.

Abbate: So new companies? People starting up?

Johnson: Usually companies that had gotten a product out there and were having some success with it but needed help with customer support. Because I had developed this reputation for having a great customer support department.

Abbate: And why was that?

Johnson: Because my philosophy on support was to eliminate the need for it. So we gave our customers all the support they needed, but we tried very hard to make sure they didn't need very much -- to make things as simple as possible so that they could solve their problems themselves. But our customers always raved about our support, because we didn't stint -- we gave them what they needed. But the goal behind all that was to make them self-sufficient so that they didn’t need a lot of support over the long haul.

Anyway, I had developed a great reputation for that, so a lot of companies would call me and ask me to come in and help fix their customer support department. Unfortunately, rarely was it a case of a bad attitude on the part of the customer support people; they almost always were trying as hard as they could. But it was either the case that the code was so complex that you had to have a really knowledgeable programmer to solve any problems, and for the most part they didn't know the application. They didn't have the accounting knowledge or whatever it was. And the person on the other end was always explaining the problem in terms of what they needed to do which the programmer didn’t understand.

Or you had a case that the customer support person knew the application and could relate to the customer’s problem, but if it was too technically complex, they couldn't debug it. Almost always the problems were systemic, not individuals. So I would end up wanting to write a recommendation that they completely rewrite their software so that a person who is only reasonably technically competent can find the problem. This was not what people wanted to hear. So you make little adjustments around the edge; you try to find ways to make it more systematic -- help people structure things so they know better what questions to ask to get the right answer from the customer so they can help them. That was the kind of thing I was doing.

I'd go in and work with their customer support people and try to understand why they were having so much difficulty helping their customers.

Abbate: Were there problems with the way software products companies were managed? Did people have difficulty figuring out the right way to manage the company?

Johnson: Well, they all had different problems. Companies that were created by people who were primarily technologists often didn't have a basic understanding of the kind of records you have to keep for a company. They didn't know basic bookkeeping and they didn't know that if you're going to assume a certain amount of cost for somebody's time -- let's say you're going to do a fixed price bid -- you're going to have to keep track of the time of the people working for you so you have a reasonable idea of how much it's going to take to do that. Things like that -- just basic record keeping for the most part. I ran into that a lot. And again, people are often resistant to imposing some system and order on what they were doing.

Or you had people who came into it from the marketing end, and they really knew how to sell, but they didn't have a good idea about how to create a quality product. They could sell it, but then once it got delivered it was just impossible to get it to work and they got really overextended because they had programmers constantly working with the customer trying to get it to work. So it always seemed to be difficult for people to find that middle ground where they knew how to sell the products, but also knew how to create an organization that would be able to support it.

Obviously many people were able to find the right mix of knowing how to sell products and also knowing how to develop and support them because over time there have been a lot of successful software companies.

One problem for so many years was that software companies weren’t able to get the financing they needed to buy the skills they didn’t already have. So they were always trying to fund their growth internally and it just takes too long. They couldn't buy the kind of skills they needed to supplement the skills they didn't have. The trick is to have enough money to hire the best of each kind of skill and then have a really good manager that knows how to make them all work together. But it takes money to do that, and when you don't have money then you're trying to cover all the ground yourself. A lot of these people had to struggle to make it all work.

I'd like to see more women entrepreneurs. That's one thing that’s changed a lot in the last thirty years. There are more women entrepreneurs now than there were. And that's true across the board – including in retail and other areas. You see many more women going into business for themselves.

As a matter of fact I read some statistic now that there's more women starting small businesses than there are men.

Abbate: Yes, I think that's true.

Johnson: I still don't think it's true in technical businesses.

Abbate: Say you want to get venture capital, there's a personal network going on there with the venture capitalists.

Johnson: That's right. You’re not in the network unless you go out and play golf with them. That's where a lot of these deals get done -- on the golf course.

People ask me why aren't there more women. I don't know, when you can have so much fun. I guess you have to have a certain level of risk tolerance. You have to be willing to face the fact that this may not work and you may be scrambling around to find something else. But women live their whole lives with risk, you know?

Abbate: I guess it depends. Were you being supported by your husband at the point when you went into business?

Johnson: Yes. That was an advantage when I started the company. He had a regular salary. And that's why I thought I would work part time. I got married the second time in 1970. Up to that point my salary was really a big deal. He had a regular salary so that really gave me a lot of flexibility.

Within two years, I was making more money than he was. He worked for a county government and had regular raises, but my salary was rapidly going up because the company was growing. That was one of the problems we had. We were pretty evenly matched at the beginning economically, and then my earnings went way up very rapidly and his stayed pretty level. That became a problem.

Abbate: I think a woman who is the sole support of her family is probably not going to take the risk.

Johnson: No, you can't. And on as simple a thing as medical benefits. If you've got kids, you've got to be in a situation where that's provided. I think the whole entrepreneurial system would change drastically if there were some way to have medical coverage at a reasonable cost without having to be someone’s employee. It's just amazing to me the way our system requires that you be employed to get good medical coverage. It impacts the ability of people to be independent and go out on their own.

Abbate: Let’s close there. Thank you very much.