Oral History of John Postley

Interviewed by:
Luanne Johnson

Recorded: March 26, 1986
Los Angeles, California

CHM Reference number: X4565.2008

Table of Contents

BACKGROUND OF MARK IV AND ACQUISTION OF AIS BY INFORMATICS....3

ORIGIN OF THE CONCEPT OF SOFTWARE AS A PRODUCT....6

ORIGIN OF ACM'S SPECIAL INTEREST GROUPS....6

ORIGIN OF THE CONCEPT OF SOFTWARE AS A PRODUCT (CONT.)....7

FUNDING OF MARK IV DEVELOPMENT....9

INTERNAL CONFLICTS ON THE SUPPORT OF MARK IV....10

COMPETITION WITH IBM'S GIS PRODUCT....11

DEVELOPING THE SOFTWARE PRODUCT BUSINESS MODEL....12

ONGOING SUPPORT FOR PROGRAM BUGS AND UPGRADES....14

FIRST SOFTWARE USERS GROUP....15

IBM UNBUNDLING AND MARK IV PRICING....16

POLICY ON WARRANTY....18

CHANGING CUSTOMER SUPPORT REQUIREMENTS....19

SOFTWARE PRODUCTS VS. PACKAGES....20

EVOLUTION OF THE TERMINOLOGY....22

EDUCATION THE MARKET ABOUT SOFTWARE AS A PRODUCT....23

INTELLECTUAL PROPERTY PROTECTION ISSUES....23

John Postley

Conducted by Luanne Johnson

Abstract: In this interview, John Postley, the developer of MARK IV, one of the earliest successful software products, describes the evolution of MARK IV from earlier programs and how the Advanced Information Systems, the company that developed the predecessor programs was acquired by Informatics. He describes how he developed the concept of software as a product and the internal conflicts within Informatics resulting from concerns that the use of MARK IV to replace custom programs would cut into Informatics' professional services business. The impact of competition from IBM's GIS [Generalized Information System] program product is discussed as is Informatics' creation of the first software users' group and Postley's role in the creation of the first ACM Special Interest Group.

Background of MARK IV and Acquisition of AIS by Informatics

Luanne Johnson: I'm interested in trying to document the beginnings of the software industry. So far I haven't been able to find anyone who has created a comprehensive overview of the things that occurred starting, say, in the late 1950s.

John Postley: I think that's right. People write about computing and they treat software as not very important.

Johnson: There are at least two books out on microcomputer software people and how that industry developed. But nothing about the people who started selling software products on mainframe computers.

Postley: I've got a couple of microcomputers, an IBM and a Compac. I use them in my consulting business. Actually, I'm semi-retired. But I do a little consulting. We have a data base system which is designed for non-expert usage. It's also used by fairly expert users in different ways. One of the ways it's used a lot is prototyping a big system that you're going to invest a whole lot money in, before you get it all screwed up and do the wrong thing. We developed a prototype which may not have every last little feature that we want. But it really tells you whether you're going in the right direction. So that was one of the things I'm working on right now. But it doesn't give me time for swimming and tennis and playing my trumpet.

Johnson: In any case, I got hooked on the idea of going back and talking to the people that have been around since the beginning and try to pull it everything together. I think there are a lot of fascinating stories to be told. And I'm really having a good time talking to everybody.

I started in the software industry in 1971 and consider myself an old-timer. But it's fun to go back and talk to people who were there before I was. I found the book on Informatics' history [*Fulfilling the Computer's Promise: The History of Informatics 1962-1982;* Richard L. Forman; Informatics; 1985] fascinating. I didn't realize that Mark IV had precedents going back so far. I thought that it was a product that began about 1966, or 1967. I didn't realize that it went all the way back to 1960 with the program GIRLS [Generalized Information Retrieval and Listing System].

Postley: Yes.

Johnson: One of the questions I asked Walt [Bauer] this morning was that in his initial prospectus for the company, before it was named Informatics, he talked about the value of proprietary programs. And I asked him if he really at that time anticipated the kind of a product that Mark IV was, an off-the-shelf product.

Postley: I'd be interested in his answer.

Johnson: Or whether he was really looking at programs which would be used as tools for customs work. And he said, well, really the latter.

Postley: I'm glad he said that.

Johnson: He said that for two years it was pretty hazy. It wasn't until they made the decision to buy AIS [Advanced Information Systems] that they began to focus on software as a product.

Postley: Well, that might be a little bit misleading on how they made that decision.

Johnson: All right, then I'd like you to tell me your version of how all that occurred.

Postley: Well, AIS was a company that was a division of another company called Hughes Dynamics.

AIS started out as a division of a company called Electrada. And that company was formed by a real promoter who didn't know anything except how to promote a company. That he knew that in spades. The day the company was formed it was listed on the American Stock Exchange. The very day. That's very difficult to do. The company had Donald Douglas on its board of directors. The guy who was head of North American at that time, I've forgotten his name. Justin Dart. I mean, really big names. And none of them ever did anything at all. They were names and they were on the board. That had a lot do with how it got listed.

We were part of the company but nobody in the company understood what we were doing. And so at some point, we were sold to Howard Hughes who was setting up another company called Hughes Dynamics. Now, this is Howard Hughes. Not Hughes Aircraft, but Hughes himself.

So we operated under that aegis for a while. Then one day I got a phone call from a guy at Hughes headquarters who said, "We're terminating the company. Fire everybody. You've got two weeks." So I said, "Well, you may not want to do that. Because we have contracts with customers and some of them are cities and states and the federal government. And they may not take kindly to the idea that Howard Hughes just walked out on a contract. They may suspect that he really had enough money to finish it. So you may not want to do that."

So he said, "Oh." Ten minutes later he called back and said, "Why don't you get AIS acquired?" I thought about that for a while. I don't remember if I called anybody else or not, but I certainly called Walt, whom I already knew. And I proposed that Informatics acquire us. It's a little vague. I don't remember the details.

But we got together and I pointed out to him I thought he could acquire the company with negative cash. Because the company had commitments that had to be fulfilled and it wasn't making a profit. That turned out to be the case and the whole thing took two weeks.

Johnson: No kidding.

Postley: From start to finish. We had to get the government contracts signed over to Informatics. It all was done in two weeks. By that time we had already developed GIRLS and Mark I. The major contract we had was for Mark II. That's how we became part of Informatics. Does that agree with what Walt said?

Origin of the Concept of Software as a Product

Johnson: Yes, that agrees. What I'm curious about is where this concept of software as a product begins.

Postley: Well, I came up with that concept. I'm not saying I'm the only one, but I did come up with it. At that time I was a pretty active member of ACM. And I decided ACM didn't fit in with my interests very much. A bunch of scientific university types. I had made every effort to get away from scientific computing. I started out working at UCLA. And then I got into Northrop and then on to Hughes Aircraft where we were going to go into the computer business in competition with IBM.

But that didn't come to pass. So I went to Rand. While I was at Rand I was working on logistics which is essentially a business application in the Air Force. And we really did the system, not just write papers like Rand usually does. We actually implemented the big inventory systems. I don't know if our software is still there but the systems are still there for the B-52 and the KC-135.

I got the idea that there was an awful lot of business-type work that was common across a lot of applications. And why did we keep writing this stuff over again? Meanwhile, at that time Cobol was being promoted as a good way of writing programs and a lot of programs were still in assembler. IBM was trying to wrestle with questions like should we have a scientific machine and a business machine? Or should we have just one?

As a Rand person I got to talk to really advanced secret planners in IBM, representing the Air Force. And I became convinced that the way to go was to develop systems that would do these basic functions at a far higher level than Cobol did.

Origin of ACM's Special Interest Groups

I tried to sell that concept in ACM and I couldn't do it. I got the idea that what we needed was a special interest group in ACM. At that time there were none. So I set up a conference at UCLA and we had over 500 people there. The major speaker was Herb Grosch who scared the hell out of me because he was traveling in the Far East and didn't arrive in LA until the day of the talk.

Out of that conference we established a Special Interest Group on Business Data Processing in the Los Angeles chapter of the ACM, which at that time was ten percent of the whole ACM.

Johnson: Do you recall what year that was?

Postley: It was probably 1959. Then with some considerable difficulty ACM finally established it as an ACM Special Interest Group and I was the first chairman of both the chapter SIG and the overall SIG. That was the start of the concept of the Special Interest Group. In fact, that's what ACM is at this point, a bunch of Special Interest Groups. Anyhow, I forgot what question I was answering.

Origin of the Concept of Software as a Product (cont.)

Johnson: Well, I was trying to figure out when you first began to get the concept of software programs as a product that could generate recurring revenue from the same set of code, not just as a tool to make customized applications for people. Of software as an economic commodity in and of itself.

Postley: Well, it's hard to figure out one point in time.

Johnson: For example, I look at the evolution of Mark IV. And GIRLS, I gather, was a very generalized program but written for a specific user.

Postley: Let me tell you about that. That's interesting. I knew some people at Douglas Aircraft. At that time, in 1960, Douglas didn't have a computer department as such or maybe it was just forming one. I don't remember. But up to then every group at Douglas did its own computer work. There was no centralized control. Each group didn't necessarily have a computer – they might have used somebody else's. But there wasn't any centralized control.

They had created all sorts of applications on the IBM 704 that incorporated files. And those files could be defined in any way. And in some unbelievable ways like look at the fourth and fifth and sixth bit positions in the 26th word and that would tell you what the format of this thing was. Really. Stuff like that.

So I convinced my friends at Douglas that we could develop a system that would retrieve information and prepare reports from any file that Douglas had. They had over 100 different unrelated file formats. And we did develop that system to retrieve reports. That's what GIRLS [Generalized Information Retrieval and Listing System] was. There was an article in *Datamation*, December 1962, on GIRLS, after it was finished.

I decided that there weren't a whole lot of people with 704s and the big thing then in terms of numbers was the 1401. So I decided we could do about the same thing for the 1401, which we did. And that was Mark I. It could retrieve and prepare reports from different unrelated file formats. It wasn't as good as GIRLS. But it was very good. It was a lot faster.

Then we made improvements in Mark II which had some features that Mark I didn't have. And then Mark III was essentially the same features but over ten times as fast.

Johnson: The way it was written up in the history book is that that then became a more specialized product in the sense it was going for a more vertical market, for the municipal government market.

Postley: No, the product was conceived as a general product.

Johnson: Oh, it was. Okay.

Postley: The customer was the guy who was vertical. And we named Mark II the Urban Management Data System.

Perhaps the final event occurred in 1963, when IBM announced the 360 with an operating system. That was a big deal. Up to then all you had was IOCS. And that's when the potential for software products became clear. I wrote a paper saying we need to write a system – this is still AIS – which did all these things much more generally and operated under the operating system. I presumed, and it turned out to be true, that everybody would use the operating system. Whereas in the 1401 you never knew what they were going to do. It was a blank screen to start with.

That was perhaps the final clincher. We started to design that system. One of the contracts we had when we were acquired by Informatics was a contract with the State of New York to design a system for the state. It really turned out to be the first specs for Mark IV. The State of New York paid for the development. They didn't buy a system, they just bought the specs. We had already used the 1401 system for many, many different applications. And, in fact, so had Douglas with GIRLS.

Johnson: It wasn't limited to a particular type of user then.

Postley: Oh, no. It was in all kinds of different departments at Douglas. I don't really know which ones took it and used it.

Johnson: Did you envision this as sort of a language, like a competitor to Cobol and RPG?

Postley: No. Well, it was a competitor to Cobol in the sense that it could be used instead of Cobol. I've always tried to maintain that it was not a language. Because it wasn't procedural. It was functional.

Johnson: So you saw this as a...

Postley: I saw this as “spell out the function”. It had a data file. You could pass that data file – everything was tape – and prepare a dozen reports of different kinds, unrelated. Except that they derived from the same data but were otherwise unrelated.

Funding of MARK IV Development

Johnson: I got the impression in reading the book that there continued to be a conflict within Informatics as to whether Mark IV was a product or a tool. Clearly Mark IV was being supported as a product, an off-the-shelf product. But there always was also a sense of it being used as a mechanism to provide services. And that as the Mark IV customer base became more widespread this became even more so. Because there was now a business providing Mark IV programmer support.

Postley: Well, to begin with Informatics declined to fund the development of Mark IV and basically told me to go away as far as that was concerned. But I said, “I'll go out and get funding.” And we did get funding of a half a million dollars from outside companies who committed to buy a system based on a set of specs. Nothing else, not one line of code. So we sold these people on it. They were all in different businesses with different uses.

Johnson: Why would they be willing to do that?

Postley: Because we convinced them that they needed this. They had common problems. They had files from which they could prepare reports, which they could update, and they wouldn't have to write programs each time they did it. They believed us. And it turned out it was true.

Then, of course, that wasn't enough money. Informatics did fund it after that. And we started making money from the beginning. So we were ahead of the game as far as financial support was concerned.

Internal Conflicts on the Support of MARK IV

Initially Informatics didn't conceive of the need to provide support. Part of the reason was we were saying this is easy to use and the users could do it themselves. Well, as the system kept getting more and more capable and complex, there were a lot more alternatives. Although any one was still easy to use, the task of remembering all of them and when to use each one and how got pretty complicated.

So we got the idea that we would provide support. But the rest of Informatics – we were a separate division – objected to that. Because we were taking their Cobol business away from them.

Johnson: Oh.

Postley: And so they said, well, if there's any Mark IV support to be done, we'll do it. Problem. They didn't know how and they never learned. Never. Ever.

Then we had the same sort of problem with the service bureau. Informatics set up a service bureau. And we said we want to sell this thing to service bureaus. They said, “No, we don't want to do that. We'll provide all the service bureau services using Mark IV.” So I said, ”What percent of the market do you anticipate getting of the service bureau market?” “Well, somewhere between 1% and 5%”. “So you're telling me that you're cutting me out of 95 percent of the market?” So I won that argument and we did sell it to service bureaus.

But as far as the support was concerned, we never really did win that argument until some people left my division, went out on their own, and set up companies to provide Mark IV support. At this point, Informatics has acquired back some of the companies and now provides extensive Mark IV support, or at least did the last I knew. What they do as part of Sterling I'm not so sure.

Johnson: Well, I'm really interested in the old days. I'm trying to envision what it must have been like for you and the other people at Informatics at that time. You really had no model to fall back on as to how to market a software product, how to support it, how you present it to the marketplace.

Postley: We used to provide free support. And I finally persuaded them... Well, I didn't persuade them, I just said, “We're not going to provide this free anymore and I don't care what you guys say.” They'd been arguing that we have an obligation to the customer. Fine, but that doesn’t mean we shouldn't get paid for it. The big problem was that there was a conflict. Because we sold Mark IV against Cobol as an alternative, not a technical competitor, but just another way of doing it. And we would say that in some jobs it would be as much as ten times easier, quicker to implement. That's where you save the time, in implementation, not in the running which is about the same.

But that's what Informatics did. They provided custom services using Cobol. So we were competing with them. So we just tried to stay away from them. But they wouldn't have anything to do with us for sort of the opposite reason. Then came IBM's GIS, Generalized Information System, which is kind of a presumptuous name.

Competition with IBM’s GIS Product

Johnson: It was probably the only three initials they hadn't used for something else.

Postley: [Laughs] That's true. And initially, they gave it away free. Certainly outside this country. Starting in 1968, I think it was, we started doing business internationally. We set up an office in Geneva and expanded from there. IBM gave away GIS. And, even though they gave it away, we never did lose out to GIS. GIS has now disappeared, but at its highest peak they had about a third the number of installations that we did with Mark IV and theirs was free.

Johnson: It was free up until the time of the separate pricing announcement, the unbundling announcement?

Postley: Yes. And it still seemed to be free in foreign countries when they felt like it, but not in the U.S. Well, they may have given it away free under the table in the U.S. though not to my knowledge. But it was definitely free in Japan.

Johnson: Some of the people I talked to who were at IBM at that time told me that their whole philosophy was to try to hold the line and only price what they really felt they absolutely had to. It was not an offensive tactic at all. It was defensive. So anyplace that they could get away with continuing to bundle they did.

Postley: Right. And I used to argue with them all the time. I had a lot of friends at IBM. When I was with Rand I used to go the IBM all the time and I'd go to private IBM parties at night. IBM doesn't drink, right? Not in the work environment. But at the parties, wow. And there'd be 50-60 people there and I would be the only person that didn't work for IBM at the whole party.

And they'd come up and ask, “What division are you with?” “Well, I'm with the Rand Corporation.”

Johnson: How serious a competitor do you really think that IBM, with “free” software products, was to independent software vendors?

Postley: Well, we won. I think that they made a big mistake. If they'd gone ahead and marketed it, like IBM can and is doing now, they would have been a very serious competitor. But they didn't. They didn't market it as a product. Even after they were selling GIS, they had it in a catalogue but they didn't really market it like they do now. They had vertical systems that they did better with. But I wasn't really concerned with those.

Developing the Software Product Business Model

Johnson: One of the things that struck me was how many of the decisions that you made back in those days when you were trying to do something that nobody else had done before ended up being a sort of standard in the way the industry operated.

Postley: Well, what I did was I took the computer itself as a model. With computer you have to have maintenance, support, upgrades and so on, so we did all that. Even the concept of the special features which was very key, very key. Are you familiar with that? We hardly ever... Only once did we increase the price of Mark IV. What we ended up doing was adding features. So if you bought the thing with all the features it was a big increase.

Johnson: But it made your market much broader because you could also go into the smaller customers with a more limited budget and sell a smaller portion of...

Postley: Actually, we didn't. That's theoretically true. But actually there weren't too many of those.

Johnson: Let's talk about the marketing for a minute. Because it strikes me that the software product marketing is so different than services marketing. Services marketing is much more geographically, regionally oriented. And you came into a company which had, I believe, an eastern, western kind of orientation.

Postley: It didn't have any marketing. In fact, Informatics prided itself that you never bought anything from a salesman.

Johnson: Oh, I see.

Postley: You bought it from the guy who was going to implement it. And he knew what he was talking about.

Johnson: This was the guy that was going to run the team that was going to do it. I see.

Postley: For MARK IV, we hired out-and-out salesmen and system engineers. We called them the same thing IBM did. Marketing representative was actually what we called them.

Johnson: So again you were using the hardware industry for a model.

Postley: Well, IBM in particular, not the rest of the industry.

Johnson: Okay.

Postley: We did produce versions for other similar machines. The RCA, whatever they called it, which was similar to an IBM machine at that time. SDS was the same way. We talked at length with Univac on the 1100 series. And we finally became convinced that unless Univac paid us or agreed to pay us up front so that we could make a profit, we wouldn't develop a system for their on spec because their users just weren't going to buy it. If Univac wanted to give it to them they might build a market that way. But we couldn't communicate with those users. It was the IBM user that we could communicate with and we did. So we modeled ourselves after IBM, everything we could, as much like IBM as we could.

Johnson: It's very unusual to have you saying in 1967, maybe even earlier, that this product was going to be the same for every customer and you were not going to customize it. Because at that time there were still people saying there's no such thing as a standard software product. Everybody needs their own unique program. So how did you get to that point?

Postley: I was convinced it didn't need to be customized. And we had some customers or prospects, Monsanto Chemical being one that comes to mind, who absolutely wouldn't buy it unless we delivered the source code. So they didn't buy it because we wouldn't deliver it. We felt that nothing good could come from that. All they could do was screw it up. Their whole concept was wrong.

Because the concept was not to modify. People then were confused and still are to some extent even on microcomputers about when you're modifying the system and when you're using the system by entering parameters or selecting from a menu or something like that. They're confused about which is programming and which is just menu-driven, to use the modern term. They couldn't separate writing code to make it do something different from selecting a module or using a module in a certain way. But the module's already there. The code is there.

Johnson: Sure, because they had never seen that done before. Everybody wrote a program to do a specific thing and they couldn't imagine that it could be written to do different things based on what you told it to do using parameters.

Postley: Where we would kill Cobol is... Some people would write a Cobol program. It was a very nice program. And it produced this output. There it is. And then somebody comes along and says, well, I'd like to have another report. So you had to modify the whole program to do a different report. In our system you could define that other report and enter it. Not one thing changed in what you'd already done up to now. Of course, assuming the same file was involved.

Johnson: Sure.

Postley: So the whole concept was that you didn't have to modify the program. Now, if we got enough pressure that it couldn't do X, that's when we came up with special features to make it do X.

Ongoing Support for Program Bugs and Upgrades

Johnson: I presume there were some cases in which there were out-and-out bugs and you had to provide some kind of a fix. When there was a case where you had to actually make a change, would you ship a new version of the object code?

Postley: You mean for the bugs?

Johnson: If there really was a bug.

Postley: We had a system of making patches. And then in the next release there would be a correction. But we didn't ship new versions.

Johnson: But you could ship out patches.

Postley: Yes. We even called them PTFs [Program Temporary Fixes] like IBM does.

Johnson: Again, the IBM model. It seems to have worked. One of the things that I'd like to hear is how you evolved your concept of customer support. For example, now with the microcomputer people they assume that they're never actually going to have any face-to-face contact with their customer and they try to provide everything through the documentation. Some of them will half-heartedly provide some telephone support but they don't do it well and don't really expect you to use it. Whereas when I first got into programming working in large shops on IBM equipment, we expected that when you bought a product from IBM or anybody else you got an awful lot of hand holding. The vendors just assumed that they were going to be involved with you to a certain point to get you running on it. In 1971 when I started my own company, I backed off a little bit from providing unlimited customer support and thought I was being very radical. But I gather that there was a decision process within Informatics as to just exactly how much you were going to provide in terms of support. Can you tell me about that?

Postley: Well, as I said before, to start with, less support was needed because the system was simpler. As we kept adding special features more support was necessary. It wasn't planned that way. It's just the way it was. So what we did to begin with is provide installation. And we would send an SE to spend a week or two and install it. I've forgotten how long, but it was something like that.

Johnson: So that was just part of the purchase price or the lease price.

Postley: Yes, that was part of the purchase price. And then they could call us up as much as they needed to. That finally became an excessive burden and didn't do the job, so we started selling support. Although the technical people kept insisting that we owed the customers the support.

Johnson: Did it become a burden because of the increasing complexity of the product? Or because of the expanding customer base or both?

First Software Users Group

Postley: Both. You'd pick off the cream first; they understood the concept a little better. What we also did was establish what I believe was the first software user group ever. That group met for the first meeting here at this house. We had I think about 35 or 40 people. My father-in-law cooked steaks on the barbecue. It was pretty informal.

Johnson: That's nice.

Postley: That was the only meeting we had here. Actually, the meeting wasn't here. The dinner was here. The meeting was down at the Bel Aire Center.

Then Informatics started giving courses. I don't remember when. But we had people who would give training. It was partly custom training and partly on-site training and partly training at a hotel, or our sites. And Mark IV training became a fairly big business on its own. That's separate from support. Support is where the customer brings you in to implement the system for them.

IBM Unbundling and MARK IV Pricing

Johnson: Did you have any feeling that there was some specific impact on your market when IBM unbundled?

Postley: Yes, we were worried about GIS. Only GIS.

Johnson: Yes, they would be a direct competitor.

Postley: Yes. I guess I can't say we never lost, but generally we didn't lose to GIS because we were better. But IBM would offer all kinds of interesting deals that we didn't always understand completely. And so there were some cases where people installed GIS, in spite of us.

Johnson: One of the things that Walter mentioned that I found interesting was the fact that with Mark IV you always worked off a price schedule. And that you had volume discounts but very rarely did you ever cut a special deal of any kind.

Postley: Right. Almost never. I can't think of a time.

Johnson: That was one of the things that I found most frustrating in the early 1970s is that there was no such thing as an advertised price on some software products. It was all negotiable. I thought that was just an appalling way to do business. I went to Indianapolis for a couple days a couple of weeks ago and went through the old ICP directories starting in 1966 up to about 1971. And, boy, almost half the products listing did not include a price. They all said that the price is negotiable.

Postley: We listed price from Day One.

Johnson: So again, I think you were very much ahead of the rest of the industry.

Postley: But, again, our model was IBM hardware, how they sold it, how they maintained it, supported it. That was the model. And that had a price schedule. What could I do?

Johnson: Talk a little bit about how you came up with the price on Mark IV in the first place. Nobody had ever priced a software product before. How did you come up with...

Postley: Well, first I came up with the price, $30,000. And I don't know how. I just decided that was the right price. And then I had to fight off Informatics. I never fought off any customers. But I had to fight off... Informatics hired a guy to be the corporate marketing VP and he came up with a price. And some of the existing VPs came up with a price. And then we hired a consultant who came up with a price. Maybe even hired two consultants.

Anyhow, I managed to prevail. I really don't know how except that they didn't have any better arguments than I did is what it boiled down to. I just felt that that was a price that somehow reflected the cost of programming that you would have to expend to kind of get going with Cobol or assembler or something like that. It had no relationship whatsoever to our cost of developing Mark IV. Some people tried to tell me it should. But I said no.

Johnson: This is really what you were saving the customer. And therefore, the customer would be willing to pay.

Postley: What we saved them initially, not over time. We never had any arguments about the price except for that first system. We came out with a really low price for additional installations to start out with. But we got rid of that fast. It was too low. So that did undergo some adjustment. But we never changed the price of the basic Mark IV system.

No, that's not true. We changed it once.

Johnson: When was that? Do you remember? It must have been in the 1970s sometime.

Postley: Yes, the 1970s sometime. We had a little advantage in Europe. We made the world's greatest, best decision. By accident, of course. We decided to price the system in Swiss francs. And the Swiss franc went through the roof. We didn’t change the price but it cost almost twice as much as it did in this country. The price didn’t change, but the Swiss franc changed – it went way up. It worked out great.

Policy on Warranty

Johnson: One of the other decisions you must have had to come up with at the time with on a warranty. Of course, this is a big issue these days now with microcomputer people and what they really are obligated to do. I think I know what you came up with but why don't you tell me in your words what the warranty really was on Mark IV?

Postley: You want to know what it really was or what it appeared to be?

Johnson: Well, both.

Postley: We felt that if we gave them a system in good faith that worked as well as we could make it work that we didn't have any further warranty obligation than that. If there was something that didn't work they could work around it or not use it. That's the way they got it and that's how it is. And basically that's the way we played it all the way through.

What we did do, however, is we finally came up with an annual maintenance agreement. And with that there was an implication – we didn't provide a written commitment – that we would deliver a new release once a year at least. And that worked out fine. People would call up and tell us about a problem. If we identified it as a bug, we would give them a workaround and say it would be fixed in the next few days or as soon as possible. Or we'd send them a patch if that was appropriate. The maintenance agreement did commit us to do that. But we didn't warrant that it was free of errors.

Every now and then you run into some lawyer whose job it is to find fault. So he does.

Johnson: Did you perceive of this as a problem?

Postley: Yes. Actually, we perceived of it as a bigger problem than it was. As a practical matter, I think ultimately customers realized that there isn't any way that you can guarantee that there aren't any errors in a system where there are literally many, many – who knows how many – millions of combinations of things that can occur and you couldn't possibly test them all.

Changing Customer Support Requirements

Johnson: Let me make a comment based on my own experience. One of the things that I observed in the years between 1971 and say 1983 was the declining quality of bug reports to us from our customers. In the early to mid-1970s, even though problems would be encountered by the end users of the accounting applications we were selling, the problems would be reported to us through the data processing department. And they would be evaluated. The DP staff would take the trouble to make sure that this really was a program bug, not a data entry or user problem, before it would come back to us. As we got onto the smaller computers and there was less of an interface with data processing people, or the data processing people were less experienced, we found that our bug processing reports became much more of a burden because we didn't have the quality control that our customers had been doing to make sure that what they were reporting to us was really an accurate report on what the problem was. Do you think to some extent your customer acceptance of the way you were handling this had to do with the fact that your customers were data processing people? That they were sensitive to the fact that there’s no such thing as a perfect program?

Postley: Yes. We certainly sold the system to the data processing people. It was not common at all that the real end user department would buy a system. I don't even think that ever occurred. If it did, it was very rare. A lot of times though we'd get reports that were incomplete or wrong. I remember one time Eastern Airlines was complaining that something didn't work. We looked at everything they gave us and we finally concluded that they must have made a change in their operating system. We told them that and we argued about that for a long time. Finally it turned out they did make a change to the operating system and the guy who was talking to us didn't know it. So he was being honest. And, you know, what can you do about that?

So there were always... In fact, this morning I was helping a guy who’s a doctor. Doesn't know anything about computers. He must have called me six or eight times. And finally, he screwed up the files that I had developed as a test for him to play around with. Fortunately, however, I could just send him a diskette with the same files on it. So he could just get rid of the screwed up files on the hard disk, bring the new ones in and start over again with no trouble. That apparently worked.

Johnson: Or else he's still figuring how to load them down to the hard disk.

Postley: He can do it. He's bright. He just doesn't know computers.

Johnson: I've been doing some consulting in dBase III because some friends of mine who are attorneys asked me to do it. I said, “Well, I don't know anything about dBase III.” They said, “Yeah, but you know computers, so you must know this.” It turned out to be pretty easy for me to pick up with my programming background. But I've been surprised at how these guys who are very, very bright, don’t know things that I just take for granted. I assume that everybody knows that a record contains so much information and you have a whole bunch of records in a file and there's a certain way of processing them. I'm finding I'm having to explain things at an entirely different level.

Postley: I have exactly the same problem.

Johnson: It's fun. I think it's really challenging. I'm enjoying trying to transfer that kind of information to people who really have no background in it at all and try to figure out the ways to describe it to leap that hurdle of somebody who hasn't had the years of programming experience.

Postley: Have you tried writing a manual? You've got to put it down on paper. That's even harder. You don't know what misconceptions they have.

Johnson: Yes, I did plenty of that for years with the accounting applications trying to write the manual for the accounting user and trying to think through all of the things that they wouldn't understand.

Postley: Do you do accounting systems for microcomputers?

Johnson: I haven't. Argonaut has stayed with the minicomputer or what used to be called the minicomputer. Now they're called the smaller mainframes or whatever.

Software Products vs. Packages

Can you go back in your memory? I'm trying to find out what companies were out there at the time. It's becoming pretty clear that Informatics was the one company that was going into software product marketing in a big way, really approaching it very aggressively. I'm trying to figure out who else was out there. We know ADR was out there with much simpler product with less function. But they were clearly marketing it as a product.

Postley: They were marketing it as a product.

Johnson: Who else can you remember that was out there?

Postley: They were the only ones that preceded us.

Johnson: There was a product called SCERT from Comress. Do you know when they showed up?

Postley: Oh, I remember that name. But that's it. I don't remember anything but the name.

Johnson: Okay. I was just trying to see if you could remember any others. I don't want to leave anybody out.

Postley: I have an advantage. By the time we got to 1960 when we actually started developing products... In the first place, I always insisted that they're called software products. Even today I insist on that in my own domain as opposed to packages. I don't like the term package. My view is that if you buy a package, it's wrapped up. It's got a string around it. Here you are. Goodbye. I don't want to see you anymore. You bought this package. That's it.

But a product is something that is treated like a product and supported and is maintained and comes out with new releases. And it's a whole different ballgame than a package. And I also felt that they have to be called *software* products. You may remember that when IBM did start selling they called them *program* products. But now they call them software products which I thought was nice of them. They don't acknowledge that I was first but they've got it right now.

I had the advantage that when I got into the product business, I had already had twelve years of experience in the software business. It wasn't called software then, I don't know what it was called. A program.

I started programming, but after that I was managing and designing. And I got into a lot of different kinds of areas. One time when I worked at Northrop I had a computer room bigger than this one with a computer in it. I was the only one who used it, the only person who used it. Except once a quarter they would send an Air Force team in for me to train. Meanwhile, I would develop and improve the system and stuff like that. I had a big bunch of people working for me. They did other things. This had to do with celestial guidance systems. That's when I got out of that business. I didn't want to be in the scientific field.

So I had that advantage of a lot of quite varied experience which is where I got the idea of software as a product in the first place. And I also had the idea that software and hardware were conceptually alike. I had this computer which was supposed to make computations for guidance paths, but I did all kinds of other things with it that had nothing to do with that. And it was the same computer. They didn't change the computer. It had its little instructions which you might say corresponded to menus.

Evolution of the Terminology

Johnson: Where did the word software come from? It just occurred to me driving over here that it might be interesting to trace the origin of that word.

Postley: I don't know.

Johnson: Clearly it was somebody trying to define part of the computer that wasn't hardware.

Postley: That's, I guess, like firmware. But I don't know who was the first to use the term or when.

Johnson: You reminded me of that thought because you mentioned that when you got into software product development, it wasn't really called software. It was called something else.

Postley: I started right down there [points out window]. I can see where I started at UCLA. I worked there and we built the first computer on the West Coast.

And we did programming for it, but it was called programming. There was no such term as software.

Johnson: I can't remember the first time I heard it either. When I started in the data processing field in the mid-1960s, what we were doing was called programming. I guess I first heard software in terms of something that you got from somebody else rather than what you created yourself. Like software that you got from IBM that was part of the operating system. That I thought of as software.

Postley: That makes sense.

Educating the Market About Software as a Product

Johnson: So I didn't know whether it was therefore the hardware manufacturers that began to use that terminology or where it came from.

If customers didn’t perceive of the concept of software, was that a market education problem?

Postley: Well, certainly one problem was that people expected to get it free. Because everything was bundled. I don't know if there are many vestiges of that problem today. But we did a lot of business in Japan. And boy, that was the expectation in spades in Japan. They expected support cradle to grave. They expected the vendors to do everything. And they did. I can't see how they made any money out of it. But they did.

Johnson: I think in terms of writing about the industry what would make it an interesting story would be to tell about the kind of obstacles which everyone had to overcome, which I think to an extent are different than they are in other industries. I started out thinking that IBM as the competitor was a big part of it. But I'm becoming convinced that that was only part of it. There was clearly a problem just in general with people believing that a program that was written by somebody else would never work in their shop.

Postley: That's right. They thought it had to be tailored to their needs. That was a big problem with our products. They'd say, “Tell me other people in the X industry who are using it.” And, we’d say, “Well, there aren't any other people in the X industry, but what is it you do that's so damn unique?”

And another problem was price. It wasn't that they were objecting to the price of $30,000 but that they really couldn't conceive of paying anything at all.

Johnson: The whole concept of paying for somebody's intellectual work.

Postley: Yes.

<u>Intellectual Property Protection Issues</u>

Johnson: Did you ever feel that protecting that intellectual work was a major problem?

Postley: Yeah, we worried about it. But we couldn't think of anything to do about it. So our solution was to quit worrying about it. We still don't have an answer to that question.

Johnson: It seemed to be generally the case within the mainframe software business that there was so much involved over and above the product itself that protecting the product wasn’t that much of a concern.

Postley: As the product became more and more complex we felt they really couldn't use it without us anyway.

Johnson: But that has certainly become a big problem for the microcomputer people.

Postley: In my opinion the number of microcomputer products that are out there being used without a license is probably 80 or 90 percent.

Johnson: The thing that surprises me so much is the I have friends who are professional people – not kids – who think nothing of making a copy of somebody else's software. They claim that if they really like it and they really want it they'll go out and buy it and get the manual and so on. But these are not unsophisticated of people who are incapable of understanding the ramifications of what they're doing. They feel somehow entitled to do that. And I think that's a real problem.

Postley: Absolutely. In fact, when you have copy protection on your system, they pretend they're objecting to the inconvenience. Some of them are. But what they really object to is that they can't make that free copy and they don't want to pay. Their attitude is that it's not costing you any more money if I use it on my computer.

In the case of Mark IV, we initially made the decision and stayed with it that we would sell it by the installation no matter how many computers they had. But if it was a different location then it would cost additional. Then you had a second installation. I remember at the time we sold it to them, the Ford Motor Company had 44 computers in one room. They paid one price.

Johnson: That’s very interesting. Well, thank you so much for spending the time with me. It’s been really enjoyable and informative.