



## **Oral History of Francis V. (Frank) Wagner**

Interviewed by:  
Luanne Johnson

Recorded: March 27, 1986  
Los Angeles, California

CHM Reference number: X4366.2008

© 2007 Computer History Museum

## **Table of Contents**

EARLY SOFTWARE PRODUCTS .....	3
STRUCTURE OF THE EARLY SOFTWARE PRODUCTS INDUSTRY .....	6
PRICING MARK IV .....	8
MAINTENANCE, CUSTOMER SUPPORT AND UPGRADES .....	9
MARK IV AS PRODUCT VERSUS TOOL .....	11
EARLY INDUSTRY CHALLENGES .....	13
PUBLIC VERSUS PRIVATE PROGRAMS .....	16
IMPACT OF IBM UNBUNDLING .....	17
PROBLEMS WITH EARLY SOFTWARE COMPANIES .....	18
SECOND WAVE OF SOFTWARE VENDORS .....	19
ACQUISITION OF PRAXA ACCOUNTING SOFTWARE PRODUCTS .....	20
ONE-PRODUCT COMPANIES .....	21

## **Francis V. (Frank) Wagner**

**Conducted by Luanne Johnson**

**Abstract:** In this interview, Frank Wagner, one of the co-founders of Informatics, describes the challenges that Informatics faced as one of the very first companies that sold a software product (the MARK IV file management system). He talks about other very early software products and the emerging understanding on the part of software vendors (and customers) of the difference between “packaging” a computer program to sell to other users and creating a true software product. The challenges that he covers are: determining an appropriate price in a marketplace where software products were not yet common; the difficulty of educating customers as to why they should buy a software product instead of developing a customized system; the internal conflicts within Informatics between the division that was charged with selling MARK IV as a product and the professional services division which saw MARK IV primarily as a tool to expedite their programming contracts; and the difficulties of investing in additional products to expand the product line.

### **Early Software Products**

**Luanne Johnson:** For my research project, I'm interviewing people who were influential in creating the initial software industry.

**Frank Wagner:** You're going to restrict yourself to the independent software vendors?

**Johnson:** Yes.

**Wagner:** You don't want to know about what the computer manufacturers did.

**Johnson:** That's the line that I've drawn around it. What we take for granted now is that it's a self-evident idea that a set of code and the related documentation and so on is a product that you can sell over and over again. When you look back, you realize that it was not self-evident at the time. It was very creative for people to come up with that concept. I'm trying to figure out

who those initial people were. And, of course, Informatics was one of the first firms that was offered software as a product.

**Wagner:** Have you seen the history of Informatics that was written by Richard Forman?

**Johnson:** Yes, you sent that to me. I haven't had a chance to read it all yet but I read the parts of it that appeared to be relevant to the product part of your business.

**Wagner:** In the chapter on MARK IV we allude to the predecessors that we could remember. There was a product that came out in parallel with MARK IV, but I don't know if anyone has ever attempted to actually document which was delivered first.

**Johnson:** There was a product in parallel with MARK IV?

**Wagner:** Yes, it was ADR's.

**Johnson:** Oh, Autoflow.

**Wagner:** Autoflow. I don't know which one was delivered first.

**Johnson:** It depends on whether you talk to John Postley [of Informatics] or Marty Goetz [of ADR].

**Wagner:** Well, that's easy to document. Because you should be able to get the copies of the purchase orders and the invoices. I think in the book I pin down a month and a year when the first MARK IV was delivered and accepted.

**Johnson:** I'll have to look at it again to pin down the month. It was announced in '67 and delivered in '68.

**Wagner:** I think it was January or something like that. It was very early in '68. I don't know when the first Autoflow was delivered. But there was one other product prior to that. It never was much of a success in the sense of a large market. But I believe it could be considered a success as a software product because it was used in the same form without modification by a number of customers. That was a linear programming program. I'm not sure what computer it was written for. I think it was the IBM 7090 or 7094. The program was developed by CEIR. It's my impression that CEIR had a contract to develop this for one oil company. Somebody in

CEIR – I have no idea of who – got the notion that, gee, let's productize this and sell it to more than one oil company. And I believe they did. Now, the person that you should try to track down to ask that question would be Bob Patrick who was working for CEIR at the time.

**Johnson:** Do you know where he is now?

**Wagner:** Last I heard he's still a consultant. And he was living up in Tehachapi somewhere. As I understand it, he lives on a ranch up in the mountains there somewhere but travels 50% of his time on his consulting and enjoys himself on his ranch the rest of the time.

**Johnson:** CEIR was...

**Wagner:** That stood for the Council for Economic and Industrial Research. It was founded by a guy named Herb Robinson as an economic consulting outfit. A fellow who might also know something about that is Herb Grosch because Herb worked for them. And I think that the chief technical man on the project was Bill Orchard Hayes. I think Grosch lives in Europe, in Switzerland. Anyway, they were started as an economics study outfit just like Planning Research Corporation was. They found out that computer programming was necessary to satisfy the needs of their clients. They started hiring programmers and then found out that you could make a lot of money by doing contract programming. And the contract programming soon was 90% of the revenues. And they were the biggest.

**Johnson:** Oh, they were?

**Wagner:** They were the biggest contract programming organization in the late '50s. But they had one fatal flaw. Herb Robinson was not a computer person. He just couldn't keep people. So they gradually disintegrated. But they may have been really the first independent software company that sold a software product.

There's one other that I have a vague recollection of. It was a performance analyzer or simulator of computer systems.

**Johnson:** Was that SCERT?

**Wagner:** Yes. SCERT was pretty early. And I have heard people claim it was as early as MARK IV but I don't believe it. But that's the only other thing I've ever heard of that claims to be one of the pioneers.

**Johnson:** Yes, that's a name that somebody else threw out. But he wasn't sure exactly when and where it was first sold.

**Wagner:** A fellow who could tell you something about it is Paul Connelly, an ex-Informatics person, who lives in San Francisco. What was the name of the company?

**Johnson:** Comress.

**Wagner:** He worked for Comress early on. He might be able to tell you what year he left them and what was the status of SCERT at that time and so on.

Those are the only possible first software products that I know of.

**Johnson:** I want to try to identify which was the first company that was formed specifically to sell software products. The companies you've mentioned all started in business to provide services of some kind. It appears that all of the early software products were byproducts of a services business.

**Wagner:** Well, I think it would be safe to say that ADR rapidly downplayed contract programming.

**Johnson:** Yes, and moved very much into a product business model.

**Wagner:** I'll take a wild guess. I'd say five years after Autoflow came out their contract programming was probably down to 25 percent of their revenues or something like that.

### **Structure of the Early Software Products Industry**

**Johnson:** I went back to Indianapolis and spent a couple of days going through Larry Welke's ICP quarterlies [directories of software products available for purchase which were published quarterly] from '66 up through about '70.

**Wagner:** They started in '66?

**Johnson:** Yes. Isn't that fascinating?

**Wagner:** I didn't know that.

**Johnson:** I didn't know that either. I thought that Larry started in about 1970. He started his newsletter to the software industry in '70. But he actually started his directories in '66. A lot of the companies listed in the first couple of years were banks who were offering software packages. One thing I found very interesting in the history of Informatics was your distinction between a product and a package. I had never heard that distinction before and it was so clearly put.

**Wagner:** Well, I tried early on to convince people to make that distinction but nobody was interested.

**Johnson:** I think that one of the reasons why there were so many things that came out that were marketed as software products that just disappeared along the way was because they really weren't products. They were packages. So a lot of the companies listing software packages in the '66 and '67 directories were banks...

**Wagner:** Simply selling what they had.

**Johnson:** To other banks.

**Wagner:** Take it and goodbye.

**Johnson:** There were some products that were being offered by companies that were contract programming firms and quite a few being offered by universities. Universities were listing what they called products but which were clearly packages under your terms. And for the most part they were selling them just for the cost of reproducing the tapes and documentation.

The first listings in the ICP directories by companies that were familiar to me as software product companies were Dick Thatcher's company, Atlantic Software, which showed up in late '68 and Lee Keet's company, turnkey systems, inc., which showed up in late '69. So I'm going to talk to both Dick and Lee at the ADAPSO Conference the first week in April, and let them battle it out between the two of them as to which one wants to claim to be the first.

The ICP directories are a pretty good source of information but they're obviously not complete. Larry traveled all over the country seeking out companies that were selling software products but he wasn't able to find everybody. But I'm hoping that as I contact people about their early

experiences in the industry, they'll know who their competitors were and I can begin to develop a pretty good list of the companies that were in the industry at the time.

**Wagner:** We didn't pay a lot of attention to competitors because MARK IV's price was way ahead of everybody else. So except for IBM's product GIS, there really weren't any competitive products. Then a fellow who had worked on MARK II or MARK III left and started a small company and put out what was basically a carbon copy of MARK IV. We worried about those kind of people. But most of the other products were priced much less. For example, Autoflow sold for \$2,000 or something like that. MARK IV was up in the \$30,000 range.

### **Pricing MARK IV**

**Johnson:** That's an interesting thing I'd like you to talk about. One of the things that struck me in reading through the history, particularly that chapter on the history of MARK IV, was how many decisions had to be made at Informatics at that time. You were out there trying to do something that had not been done before. And many of the decisions that were made became standard to the industry. One of the things that fascinated me was the discussion on the pricing, how they came up with the price. There was no existing business model to use for establishing the price of a major software product. What memories do you have of that process?

**Wagner:** Well, we hired consultants to do a so-called market study. Which was very difficult. Because when you asked anybody did he need it, would he use it, it was black and white. If he knew about the predecessors to MARK IV -- the work up at Hanford, or Charlie Bachman's 9PAC done under SHARE auspices, or IBM's military FFS, Formatted File System -- he'd understand what it could be used for. But there were very, very few data processing managers that knew about any of them. Maybe a dozen or twenty. And we already knew who they were so we knew what they would say. Anyone else that you went to, you had to explain to them what a file management system is. And you had a whole educational process to go through before you could ask them: Would you buy one? What do you think you'd pay? And so on.

But in any event we still tried to do it. We hired Don Dickburner, whose company I guess is still in existence, Dickburner and Associates. He did some face-to-face interviews and a lot of telephone interviews and came up with numbers which said if the price is this much over the next five years, you'll probably sell this many. If the price is twice that, you'll sell half, or whatever. And created a supply/demand table.

**Johnson:** Which is interesting as opposed to a typical manufacturing approach where you price it based on what you expect it to cost you.

**Wagner:** And how many you would have to sell to get your money back.

**Johnson:** I found that fascinating.

**Wagner:** Now, my approach was very simple. I started with the fact that suppose you really want to place a lot of them, you've got to rent them. If you're going to rent them, the monthly rental should be equal to the monthly rental cost of a tape unit. Why? Because a data processing manager can authorize and order rental of a new tape unit without going any higher than himself. So therefore, he can authorize renting a piece of software for the same price without asking his boss and his boss' boss and getting the purchasing department into it and all that which would kill the sale.

**Johnson:** And so was that policy implemented?

**Wagner:** Well, almost. And then take that and multiply it by some factor, maybe forty, and that's the perpetual licensing fee. So what it finally came down to is that Postley set the number. We came up with numbers from \$15,000 to \$30,000. Postley said the introductory price will be \$25,000.

**Johnson:** I remember Larry Welke telling me a story once about research he did in the early '70s asking various software vendors as to the logic behind their pricing policies. Because the prices were just all over the board. One guy told him he had priced his product based on the cost of a Lamborghini. Because when he sold the first one he was going to buy himself a Lamborghini. A lot of scientific research went into the pricing in those days I can tell!

The other thing that I thought was very interesting was that initially Informatics provided maintenance for free.

### **Maintenance, Customer Support and Upgrades**

**Wagner:** That's right. There was a big difference of opinion on that. Postley's point was, yes, we should charge for it but to capture market share, I'm going to give it away because that will enable me to capture a bigger share of the market faster. And we all said, oh, you're making a big mistake. Because when you introduce it later you'll get tremendous customer resistance. The MARK IV maintenance fee was always underpriced relative to nearly all other products. So I don't know if anyone can say which was the right decision.

**Johnson:** I thought that was a really interesting aspect of the evolution of software products. As quite a few products began appearing in the market in the early '70s, the standard seemed to be to charge for maintenance. But also with each purchase, there was an assumption that there was an embedded amount of training or support...

**Wagner:** Of training or support, yes.

**Johnson:** ...in the initial license fee. I think it's interesting that now with microcomputer software, of course, it's gone completely the opposite. The assumption is that...

**Wagner:** You get nothing.

**Johnson:** And attempts have been made to deal with that problem with documentation and with training disks and so on.

**Wagner:** Well, another interesting aspect of MARK IV was Postley's concept of the after-market, of selling special features. That was really I thought very, very good marketing. Because special features, and eventually education, became a substantial amount of income.

**Johnson:** As I understand it, that concept not only provided a way to market to the existing customer base, but it gave you an entry with each new customer. Because you could always sell just a portion of the total product. A lot of software companies never really understood that concept.

**Wagner:** And then next year sell them additional features. Well, you know, in its heyday I guess a full blown MARK IV with all the bells and whistles might have cost \$120,000. But you'd sell \$50,000 worth and then in the next two or three years you'd sell another thirty or forty thousand dollars worth. And that selling was very inexpensive. In fact, it was frequently done by the SEs [System Engineers who provided customer support] not the salesmen.

That was another thing which I never really fully understood about how all the other software companies operated: the number of field technical people who were not commissioned salesmen in ratio to the number of commissioned salesmen. We always seemed to have a lot of those, and provided a lot of free support in the sense of having people ready to answer questions. At least through the growth period of MARK IV. I guess as it began to decline we cut way back on that. But that would be an interesting comparison to make at some point.

**Johnson:** It seems to me that it would have been very easy to have gone the other way, to have gone to the model of the microcomputer software companies now. To have initially said, we're going to send you a bunch of source code. You do what you want to with it.

**Wagner:** We were very much concerned about – possibly overly concerned about – the fact that we were really creating a market. And we weren't ever sure that people would buy unless we did for them the things that we as former customers needed. Maybe we were a little different from some of the other companies in the sense that all the management of Informatics had been data processing managers. With the exception of Werner Frank, we were not techies, not programmers. And so we kept thinking how would I react if a salesman came into my office with something brand new? What would I want from him? What would persuade me that this is the way to go? And so we kept thinking in terms of technical support and so on. Now, the only thing that we never dropped standards on was modifications. Postley was absolutely adamant. I will never customize. Never, ever, ever, ever. And, of course, Werner thought he was the world's most horrible salesman. Werner was in those days your classic custom salesman. He sold to the government. And if the government said I want you to deliver 400 programmers with nothing but their shorts on tomorrow morning, Werner would have said, no problem.

**Johnson:** The only question is the price.

**Wagner:** And so he and Postley used to have violent fights. Werner was convinced that Postley knew absolutely nothing about selling, nothing about marketing. He was the world's worst marketeer. And, of course, the feeling was mutual. Walter [Bauer] leaned in Werner's direction a lot.

### **MARK IV as Product Versus Tool**

**Johnson:** There was an interesting conflict reflected in the history. It was never explicitly stated but there sort of seemed to be an internal conflict as to whether MARK IV was a product per se in and of itself or whether it was a tool to be used to provide custom programming in a more effective, efficient way.

**Wagner:** Well, Walter's position was that it's a tool. It's going to be sold as a product. That's great. We'll just have to keep the cost of selling down so we can make as much profit as possible. But once having taken that position, he also took the position, now, let's exploit it. Let's find some synergy in it. Let's let the professional services group use it to deliver custom programs faster and better.

The outcome of that is that the salesman was telling the guy that MARK IV is so easy to program that your dumbest programmer can do all your work and you can fire all the high-priced people. And then the professional services people would come in and they'd say, well, now that you've bought MARK IV you've got to buy our high-priced professional programmers to get anything useful out of it. And the truth of the matter was that both statements were true. If the application was relatively simple, the less you knew about programming, the better MARK IV programmer you were. You just followed the rules and tomorrow afternoon you had the application done. But if you did a big complex job like a first-class general ledger, you had to have the right expert MARK IV people. And again, the less they knew about COBOL the better. Because they kept trying to apply COBOL concepts to MARK IV and that was wrong. But your average buyer of MARK IV just hadn't developed that level of competence. So there was truth on both sides of this story.

**Johnson:** You know, you could be talking about one of the sophisticated data base management systems that are being sold right now for the personal computers. The way they sell them is that anybody can use this to set up anything they want. And as long as you've got a very simple application, no problem whatsoever. But if you want to do anything very sophisticated you really have to get somebody who is, in essence, a programmer to do it. So that's fascinating that 20 years later it's the same.

**Wagner:** The same story. So you say was it successful? Partially. We developed a cadre of MARK IV experts on the East Coast under a fellow named Jack Gelione and on the West Coast under Tony Lamia. And working for Gelione was a fellow named Neff. We developed about \$2 million a year worth of programming in MARK IV. And it was slowly being built up. But there was a fundamental management question. Should this work be under the MARK IV organization or not? And there were pros and cons as you can imagine on that. It takes a great deal of different talent and know-how to develop business and grow business in the custom programming business versus product business. And the decision was consistently made that we shouldn't dilute the efforts of the product business. Well, the net result then was there was not good cooperation between the two organizations. That was one problem.

The other problem happened when we acquired Programming Methods, Inc. We had MARK IV programmers working for The Equitable. And Programming Methods had COBOL programmers working for them. We had a big fight over this. Werner argued that this confuses the customer when we have two different salesmen coming in and selling to them. And that we've got to transfer all these MARK IV programmers into PMI. Which we did. Over my strong objections. Within four months they had all quit because they didn't like the PMI people. I knew they wouldn't like the PMI people. So they all quit and we lost \$2 million worth of business. And Neff went off and formed a company and carried on the work. And then he hired Tony Lamia out here as his West Cost Vice President. And eventually I think Informatics bought them about a year or two or three ago.

## Early Industry Challenges

**Johnson:** Well, that's interesting. What goes around comes around, doesn't it? Step outside your role at Informatics for a while. I'd like to go into your role with ACM [Association of Computing Machinery]. You had a position of being an industry observer as well as being actively involved with Informatics. This project started because of our assumption that when IBM started unbundling or separately pricing software that suddenly there was this great opportunity out there. What I'd like to try to zero in on are what were some of the problems facing this fledgling industry in the '60s, IBM competition being one of the problems, perhaps. But there were also clearly a lot of other problems. There were the problems of availability of money for R&D to develop products. There was the problem of lack of skilled people out there. There was the problem of customers accepting the idea that buying a product could fulfill their needs. Could you kind of talk about that for a little bit and put it into perspective?

**Wagner:** Well, the last one was the biggest problem.

**Johnson:** You think that was the biggest?

**Wagner:** Oh, yes. Creating the market, educating the customers that a product was good for them. And you had all kinds of people objecting. You had it from a purely, purely selfish, political point of view.

**Johnson:** Was that overt, do you think?

**Wagner:** Yes, in a few cases. And there were a lot of people who would never admit it. But you knew in your heart that they had the attitude: so a product's going to save me ten man years of development. Geez, that means my empire is ten man years smaller. The hell with that. There was the legitimate fear that ran along the lines of, gee, my business is very dynamic. Things change. We acquire a company and we have to change all the chart of accounts. The government puts out this regulation and we have to change our accounting procedures. How can a rigid product handle these changes?

In some cases that was legitimate. But most of that was the worry that said I'm different. Even if you tell them, look, three of your competitors, Texaco and Gulf and Union Oil, have bought this product and are using it, why can't you, Mobil, use it? Well, Mobil's different. It's so ridiculous. But that was just educating the world to the idea they should use a product. I think that was the biggest problem.

The companies that were selling languages like MARK IV had the problem in spades. Because a lot of customers still hadn't converted from assembly language to COBOL. And then to tell them here's a language that's going to be better than COBOL, that was really tough. The software companies that had it the easiest were the ones who sold utility programs. You know, a sort is a sort.

**Johnson:** It doesn't matter whether you're sorting gas company records or retail records. It's a sort.

**Wagner:** Yes. DASD [Direct Access Storage Device] management software – those companies had it easy.

The problem of deciding how much to spend on R&D and new products is a tremendous problem that really didn't hit until the industry was down the road a ways. Because virtually every company was a one-product company. And they rode that product.

**Johnson:** And usually that product had been developed through some kind of sponsorship, direct or indirect.

**Wagner:** In many cases, yes. So starting from scratch saying, what shall our next product be, was tough. I don't think any of them understood the fundamentals of product planning. I certainly know Informatics didn't. To this day I don't think either Bauer or Postley or Frank understands how a manufacturing company does product planning. They have a separate organization that doesn't have an ax to grind, that studies in a cold-blooded way the pros and cons of all the candidates. I think most companies come up with two or three candidates at best and rapidly home in on one because intuitively they feel this is it. And then go ahead and try and develop it.

So that problem of deciding how much to spend on R&D for new product was a big one. An interesting Ph.D. thesis would be to study how much money was wasted on new products in the software industry, how many new products failed, and how much money went down the drain. I know Informatics spent a fortune.

**Johnson:** That would be fascinating wouldn't it? And it's always amazed me how many companies did really well with one product. And then the obvious thing is to build up the product line and it is rarely successful.

**Wagner:** Now, in your line, in the applications products, particularly accounting products, it was fairly straightforward. If you were only selling general ledger you were at a disadvantage from a guy who could also sell you an accounts receivable and an accounts payable. So you had to have a reasonably full product line. Informatics tripped all over its feet trying to do that.

**Johnson:** Well, the product that we started with was the payroll product. I had acquired it from a defunct company, and it was beautifully written, beautifully conceived by just a couple of really bright guys who had done it specifically for a service bureau environment. The idea was generalized from the beginning. They set out not with any foresight but just by happy coincidence to generalize it from the beginning. And it was a really strong product and still is to this day with all its various permutations. As a matter of fact, there is COBOL code in the part of it that does the payroll tax calculation that I am sure is exactly the same as it was written in 1969. Because they really tried to generalize it and they were a couple of really bright guys. Except that they didn't know anything about running a business which is why they went defunct and why I ended up with it. I thought it would be the easiest thing in the world to start adding more products to our product line but somehow it just isn't that easy to do.

I think the main reason was, at a really small company, if I had a couple of people developing a new product, it was really hard to keep that the priority when we had to suddenly deal with a problem that occurred in our established product. That was where our revenues were coming from. If we had a problem of some kind because of a bug or a change in an IRS regulation, that became a higher priority than the development work. I know now looking back there were problems with capitalization. We didn't have enough money. If I'd had more money, I could have handled it better. But I was constantly shifting priorities.

**Wagner:** Yes, but would the money have been well invested, wisely invested?

**Johnson:** I don't know. I do know that I had difficulty maintaining priorities because I didn't have enough money. I had limited resources. And so I shifted priorities all the time.

**Wagner:** A solution to that is don't ever develop a new product. Only buy it after it's been developed.

**Johnson:** Which was my second alternative. I gave up developing. I found a guy that had a service bureau up in Canada that had developed accounting products and I bought the second, third and fourth products from him. Unfortunately, they didn't turn out to be as good as I thought.

### Public Versus Private Programs

**Wagner:** The problem with buying a product from a service bureau is that it turns out not to be a "public" product but what I call a "private" product. The father of it and the mother of it are there to pick it up and change its diaper whenever it gets in trouble. They claim they don't, but they do.

**Johnson:** But they do, yes. This ties right in with an interesting session I had earlier today with the people at BMDP Software. I learned about them from a friend who knew that there was a statistical package being offered out of UCLA in the '60s and that a product with the same name is still advertising today in PC World. When I did my research at ICP, I looked for the product in the directories and, sure enough, it was listed in one of the very early issues of the ICP Directory. So I called them up and said that I'd like to come and talk to them. There's a man named Dr. Wil Dixon who was involved with this product at UCLA.

**Wagner:** Is he still there?

**Johnson:** Yes. Three years ago they took it out of UCLA to make it a private company. It was funded originally by the National Institute of Health and UCLA and some other organization. So they were a not-for-profit organization. According to the listing in the ICP directory, they were selling it just for the cost of reproducing the tapes and the documentation.

But he did have the concept of productizing it very early on. By the late '70s, there had been ten upgrades. There were consistent releases. There was documentation. There was a limited amount of support, not a lot. But it was there. And when they moved into the private marketplace, all they really had to add was marketing. They had developed everything else. He made a big pitch for the fact that they were the first software product vendor out there. I said, no, because you didn't form a company until much later. He said that they were in essence a company, they just were within the university.

And I said, no, if they didn't have a marketing strategy, as far as I was concerned, they weren't a company. Because they just did it by word of mouth.

**Wagner:** I think it was also distributed through SHARE.

**Johnson:** It might have been. That sounds very consistent. But one of the things that he mentioned -- this goes back to this mother and father thing -- was that at one point in the late '60s to early '70s, there were about 100 other similar packages that were available free for the asking or for the cost of reproduction. And they all just disappeared. Because the person who initially started it would leave and go on to something else. And there was no way anybody else could pick it up and maintain it. So this is really helping me home in on my definition of what a

product is. You've distinguished between private and public. Obviously a public product is one that can be carried on independent of the person who conceived and wrote the code in the first place.

**Wagner:** Well, I used those terms with regard to any program. You can have a public program or a private program. But just because it's a public program doesn't make it a product. That's one of the requirements but you have to have lots more.

### **Impact of IBM Unbundling**

**Johnson:** What difference do you think the IBM unbundling really made? That's one of the things I'm trying to understand.

**Wagner:** Oh, it helped tremendously in the fundamental problem of market education. The problem of educating the market that you can really buy a software product and use it profitably was given a tremendous boost by the fact that IBM said program products are products. Now, the fact that they were talking primarily about operating systems should have logically meant that the impact should not have been that great. But that was coupled with the simple fact that software costs money. It's not free. That was a big help in education.

On the day of the IBM announcement when all the reporters and the stock analysts were calling, Walt Bauer was the only software company president that went on record and said that this will be a tremendous help to the independent software industry. It has legitimized the idea of selling a software product, of charging good hard money for it. Part of the market education problem was there were still people who felt that it was somehow immoral to charge for software, that software was really an intellectual product that belonged in the public domain and should be given away to everybody. That was the university influence. It didn't bear any logical scrutiny.

**Johnson:** Except that we are in a society that believes that information is something that should be readily available. We have libraries where we can go and check out books at no cost.

**Wagner:** That was part of educating the market, to wean people away from that kind of thinking. And certainly when IBM said software costs money and we're going to make a profit selling it that made a difference. They didn't say that on Day One but it was clear to all thinking data processing people that the time was going to come when software would be a substantial revenue producer for them.

**Johnson:** I remember something you said to me years ago, Frank. I was still very naive and just getting involved in ADAPSO and you made the statement to me that the time will come when we sell the software and bundle the hardware for free. And I thought, oh, this man must be crazy. But, you know, you're quite right.

**Wagner:** Well, it hasn't happened yet. But it can happen. I still believe that the day is coming in the next 25 years when you will buy a personal company not by buying a computer but by looking in a catalog for your application. And if your application is bee keeper, under bee keeper will be 27 different kind of products. The user's manual for the application will be the description of the product. You'll buy it and it will come in a box. And in the box is going to be the microcomputer. But you're buying the application. You're not buying the computer. I think that's going to happen.

**Johnson:** Well, you're on record now as saying it.

**Wagner:** And if you bought it from a reputable company and you bought Model 1 and three years later you find that you've exceeded the capacity of Model 1, you'll call up and say, what's the trade-in value of my Model 1? Because I now need one of your Model 3's. That's the way I see the personal computer business going. So in a sense that's selling the software and bundling in the hardware.

**Johnson:** Sure. It's selling the solution is really what it is. They all say they're doing that. That's the biggest cliché in advertising both hardware and software: we're selling solutions. But we all know that it really isn't there yet.

### **Problems with Early Software Companies**

**Wagner:** And that will solve the marketing problem. Everything will be bought out of catalogs. Now, getting back to the earlier days of software companies, it seemed to me that a great deal of the early problems, why so many companies went under, was the obvious cliché. They were run by programmers who didn't understand how to run a business. And who thought small. They didn't really think in terms of growing. They didn't think in terms of setting up an accounting system that could handle growth. If they had revenues of a million dollars a year, they wouldn't dream of setting up an accounting system that could handle revenues of \$20 million a year. And didn't think of public relations enough. Of course, it's so hard now. But when I'm talking back in the '70s, I don't think enough of the companies thought in terms of good public relations.

By good public relations I don't mean spending a million dollars on advertising. I mean, placing articles in *Datamation* about your product and appearing on the platform of every professional society meeting that you can worm your way onto the platform of. And so on. I think a lot of them didn't think big enough. They had the notion that this is the greatest product that ever was because I'm the programmer that programmed it. And people will just flock to me to buy it.

**Johnson:** It goes back to the package vs. product problem. I started my company in 1971. And one of the things that I found was that I was initially selling against two things. I was selling against a conviction the company had that they could do it better themselves or against a disastrous experience they'd had with a previous package. And that's a real tough thing to sell against. I remember feeling very elated in about 1975 when I suddenly discovered that I was in a situation where I was selling against good competitors. And I said, this is great. I'd much rather go in and sell on a feature-by-feature basis against good competitors than sell against these bad competitors...

**Wagner:** Who ruin the market. You know, in the early years, 1960 to '65, the same problem existed with professional services.

**Johnson:** Oh, really?

**Wagner:** And CEIR was one of the big offenders. You'd call on somebody and they'd, say, oh, we hired some programmers from CEIR last year and it was a disaster. We'll never again hire an outside programmer. So I can imagine how the same thing would be true of products.

Who have you discovered to be... if I define the first wave of software companies as being Informatics and ADR, what was the second wave then? You say Thatcher, Atlantic Software.

### **Second Wave of Software Vendors**

**Johnson:** Yes. I think so. And Lee Keet with turnkey systems, inc.

**Wagner:** What was his product? Was it a telecommunications monitor?

**Johnson:** Yes, TASKMASTER. Then I think we should include Pansophic with PANVALET. And Cincom came in with their TOTAL database about that time. It's so important to save the history of these companies. We have this invisible industry which sits in between the hardware industry and the microcomputer software industry of people who were doing very

significant things and dealing with incredible problems, maybe the most incredible problem being that nobody really understood what these guys were selling.

**Wagner:** In the Informatics history, there's a quote from the prospectus for Informatics in 1962 in which Bauer speaks about proprietary products.

**Johnson:** I pinned him down on that. Did he mean products in the sense that we now think of software products? Or did he mean tools to be used for custom programming? And he admitted that between '62 and '64, it was very hazy. It really wasn't until Informatics acquired AIS [Advanced Information Systems] which brought John Postley into the company that they really began to focus. But it was very clear that what he had in mind was not necessarily selling products per se. It was a matter of helping to overcome the labor shortage by finding ways of maximizing work that had been done and not doing it over and over again.

**Wagner:** What about John Imlay? Have you talked with him?

**Johnson:** No, I haven't. I'm going to talk to him in Houston. Imlay's interesting. Imlay comes into it fairly late.

**Wagner:** Does he?

**Johnson:** MSA was initially a contract programming firm. They were listing packages in '68, in '69, in the ICP quarterlies that they had developed for their contract programming customers. When Imlay came in in '71, he changed their focus. I want to pin John down as to how much of that was forward-thinking strategy and how much of it was defensive. Because they were losing so much money...

**Wagner:** Well, they went bankrupt, right?

**Johnson:** Yes, they were in Chapter XI. And I think John will be very candid as to how much of it was foresight and how much of it was defensive.

### Acquisition of Praxa Accounting Software Products

**Wagner:** I don't know if you know that we had a disastrous experience with accounting products written in MARK IV that we bought from Praxa.

**Johnson:** Was that Art Kramer's company?

**Wagner:** Oh, yes. And we nearly had a big lawsuit with him.

**Johnson:** No kidding?

**Wagner:** Yes.

**Johnson:** Oh, I never knew that.

**Wagner:** Kramer and Informatics were treating one another with icy politeness for a couple of years.

**Johnson:** It sounds like a very logical extension of what MARK IV was doing.

**Wagner:** The problem was, as I say, the products were pure private products. Once you tried to let them stand on their own two feet, they completely fell apart. And I fault our people for not understanding that when they first reviewed the product. But that was the big, big, biggest flaw in Informatics was the fact that we would look at everything through rose-colored glasses product-wise. And say, oh, this is great. It's going to sell like mad. I think it might even be true that any company really can only have one big major success in any given era, and an era is how long one man is the leader.

### **One-Product Companies**

**Johnson:** What you're saying is that one individual is capable of one great vision.

**Wagner:** One great vision. And he can make a big success. Postley and MARK IV is the thing that puts this notion in my head. But there are other examples.

**Johnson:** Isn't that really a very pessimistic attitude that each of us is capable of accomplishing one wonderful thing?

**Wagner:** Maybe it's only the programmers that are that way. The other people aren't.

**Johnson:** Well, maybe. It seems to be bearing out in the microcomputer business. They come out there with one great fantastic innovative product. And...

**Wagner:** That's the end of it.

**Johnson:** Well, that's something for me to think about to end the interview, Frank. Thanks so much for taking the time to talk to me. It's really important to make sure that your recollections of the beginning of this industry are preserved.