

DATA SYSTEMS NEWS

A SUBSIDIARY OF MEDIA HORIZONS, INC. 200 MADISON AVENUE NEW YORK, N.Y. 10016 AREA CODE 212: 686-3100

3/18/70

Recd 3/30/70

Dear Sir:

Enclosed is a transcription of a recent DSN Software Forum in which you participated.

Would you be so kind as to read your comments and then let us know if there are any adjustments by March 23rd at the latest.

The Forum will appear in our April Issue. Many thanks again for attending.

Sincerely,
Walter Presnick

Walter Presnick
Associate Editor

In attendance were: JERRY DREYER, executive vice president of
ADAPSO, New York, N.Y.; HARRY ROWELL, director of COSMIC, Athens, Ga.;
GORDON FRANK, president of Software Sciences Corp., New York, N.Y.; DAVID
AHLERS, vice president, director of management science, Bankers Trust Co.,
New York, N.Y.; JERRY LOWRIE, director of automation, The American Bankers
Assoc., New York, N.Y.; REX McWILLIAMS, chairman of the board, managing
director of National Computer Analysts, Princeton, N.J.; MARTIN GOETZ,
director of proprietary software, Applied Data Research, Princeton, N.J.;
LAWRENCE SCHOENBERG, president of AGS Computers, New York, N.Y.; WILLIAM
GRINKER, executive vice president of the Boston Computer Group, Boston,
Mass.; and Lawrence WELKE, president of International Computer Programs,
Indianapolis, Ind.

CHRISTIAN: Is the software business a ~~service~~ industry or a product industry?

In which direction is it moving today?

GRINKER: I don't proport to be a technologist. But is the programming art any different than the creation of music? You create music for a purpose and you create it for a device to play it. You create software for a purpose and it plays on a ~~computer~~. Can music be standardized? I don't think so, and I don't think programming can be either.

GOETZ: I completely disagree with the last point. Unfortunately, programming today is an art. There are many similarities, for example, between building hardware and building software. You can modular~~ize~~ software. You can use packages which can do much of the basic programming requirements, but unfortunately most programming today is undisciplined, and is an art when it really should be a science. It can be a science, and ~~unfortunately~~ I think proprietary packages may help in having that come about.

Standardized procedures might be one major advance in programming which unfortunately we currently do not have.

ROWELL: I'm a middle-of-the roader on this paticular question. We're moving toward the same outcome the architectural industry has moved toward. It will come after certain sets of specifications and standards are arrived at. Since my interest is ⁱⁿ packages, in contrast to the creation of original software, I'd have to think that way, otherwise I could not do my particular job. I think that given time, and I have no idea what this time would be, it will become structured to the point where it is cont~~rollable~~rollable. We will be working according to, and by a set of specifications that people will accept across the board. The movement toward that comes with the recognition of software as a package.

more

LOWRIE: Dave Ahlers is with Bankers Trust. He is a member of the Automation Committee of the ABA, by which we have our automation program administered and given direction ^{through} ~~by~~ a committee of bankers. Dave is the author a series of programs on investment analysis language, which the ABA is distributing through time sharing vendors to the banking industry. We feel we are a catalyst in getting this type of language to the banks. We did have, through the program of Automation Technology Research, a project known as the banking language. Right now that's in a very static situation because of the formidable job of getting any kind of consensus about the need for it and the ability to handle it from a central source. It's dormant, but among a number of bankers there's a great deal of interest in some type of standardization. ~~Right Dave~~ I think Dave, in his effort with investment analysis language, is going to give us at least some initial feeling for the use of the standard language of banking.

AHLERS: I'd like to talk about the "how" part of your question. I find it a bit incredible that we should be in such an art, making differential equations systems work on computers. I feel that many of our efforts on evaluation are in the wrong framework. A lot of people are talking about it, so let's evaluate it. Let's see how many seconds it takes to design this system, or how fast the system runs. We're trying to put the evaluation scheme in the same context in which we used to put coding schemes. I think that's just as much a mistake as saying, "oh, it's like composing music and somehow we can't evaluate it."

If you want a further suggestion as to how we might do it, I recommend the discipline of psychology.

WELKE: Isn't it true that if we knew what we wanted to get out of a programmer, we'd be able to measure what we ^{were} ~~are~~ getting from him? This is one of the basic problems. It sounds ~~simplified~~, but it would seem that if we really knew how

(more)

to define the problem in some drastic manner, we'd be able to measure much more easily what we were getting from him.

AHLERS: I agree conceptually, but the industry that I'm in is heavily data processing oriented. ~~The market~~ ^{amaunyat} The market and security clearance is not that complicated. What I ~~really~~ mean is that there are certain specifics that we can talk about. We can discuss how long it takes to get a system done, where you double over your plan, or if you delayed other systems. What about the ratio of maintenance costs to development costs? What I'm ~~really~~ saying is compatible with what was presented earlier, but this is a benchmark, otherwise we have no idea of where we are now. Our biggest problem is not in data processing our bank, but it's in management systems. The hard part is not in designing the system of the future, but in somehow evaluating where we are now. Unless we somehow come to grips right now with evaluating what is a good systems design and a bad systems design in some operational way, everything we propose will simply be compared ~~to~~ against a marshmallow, and there'll be no way of evaluating our proposal.

WIEKE: The question that keeps popping into my mind is if the independent software firms are to make a profit, it would seem that the profit has to come by being able to properly estimate the size of a job before they go in and undertake it. Obviously, some of them have been ~~profitable~~ profitable. Maybe its accidental, but I don't think the entire industry can use accident as an excuse for making profits. They must be doing something right as far as estimating what it takes to turn out a job.

GOETZ: My company is spending most of its time in proprietary software, which is somewhat different from professional services. In professional

4

services you ~~can~~ really cannot give fixed price contracts. A good part of the problem is in defining what the customer wants. He generally doesn't know, or even if he does know, the problem continues to change. I don't think the success of the companies is in ~~in~~ their ability to accurately predict their costs, but in understanding the problem and taking on only jobs which they are qualified to do.

SCHOENBERG: I'd basically have to agree with you. But the answer to how most ~~companies~~ companies are profitable is that they charge on a time and materials basis. The problem here is that when people talk about other disciplines, they try to think of the problem as something that's isolated from other interfaces. The big problem in the contract ~~and~~ programming business is that it is ridiculous to talk about your ability to standardize or to control when the element of the job you are performing is a relatively small part of an overall picture. It is not realistic to standardize or control when we're not the controlling element. ~~o~~

GRINKER: I recognize, or at least detect the fact, that you're suggesting that the software industry as a supplier is dealing in the delivery of satisfaction. Can programming and systems analysis ever become standardized to the point of being controllable? Of course, they are two entirely different disciplines. ~~System~~ Systems analysis is really delivering satisfaction to the client, while programming, on the other hand, is a way of producing a product. It certainly is a very convenient thing for the supplier if he's organized, with a large bunch of guys, to standardize his process. But it really doesn't have an awful lot to do with the result, except the manner in which he creates it.

FRANK: What you have here is a kind of a melting of two considerations. One is the creative ability of people to respond to systems and programming
(more.)

situations. The other is the pressures of business in a real-world environment to know exactly what ^{directions} they're headed in, and to have some yardstick for measuring these directions. I would think that the challenge for data processing is the need for more standardization, and to do this without necessarily killing the capabilities of people in systems and programming. We've already begun to see a general movement towards standardization, because most users realize that merely the lack of project control can kill a project no matter how many creative, competent people are on that project.

~~McWILLIAMS xxx Gordon~~

McWILLIAMS: Gordon just mentioned that standardization is evolving and that we will proceed to standardization in various aspects of the software industry. I think it would do us all a ~~little~~ bit of good to see where the most significant efforts in standardization have taken place, apart from those that are strictly industry-related. ~~And I think it would do~~

~~us a little bit of good to see where the most significant efforts~~
 And I think the most significant ones have occurred in the areas of standardized coding. We aren't all standardized there, but that's been one of the most significant attempts toward standardization. Then we proceed to attempts at standardizing programming languages, presumably those that are problem-oriented. And we only have to point out Fortran and Cobol. Fortran was a de facto standard long before it became a formal standard. But the fact is that standardization in these two programming language areas has significantly contributed to the ability to skirt the problem. The question of how programming systems analysis will ever become standardized to the point of being controllable⁽¹⁾ is murky. I think one point, and Gordon did touch on it, that's overlooked is that software development is controlled much better, and is, in fact, much more controllable than it was a few years ago, if only because we have
 (more)

a reservoir of experience that's approaching twenty years.

CHRISTIAN: ~~This~~ What do you think accounts for the lack of standardization so far in this ~~computer~~ industry? Is it rapid growth of the industry and the various approaches to programming or the competitive element?

GRINKER: I don't think it's either? I think the two things that really account for the diversity is first of all the diversity of devices. You have a lot of sources and a lot of destinations. You may have 200,000 programmers in the United States, therefore you have 200,000 ways of travelling the road between the device and the purpose. How can you possibly standardize the middle process when the source and the destination are diverse? You can organize and control the process to an extent, but the entire industry, if you will, or the programming business, is really at the mercy of the hardware producer. Software is created as the whole software business is a function of the lack of completeness of the product to begin with. So it exists at the will of the hardware device. A dramatic change in the nature of hardware could completely eliminate the requirement for software.

GOETZ: I would place the blame ~~with~~ ^{on} the computer manufacturers who, in the ~~existing~~ ^{essentially} sixties, essentially controlled all the software and directed the users on how to use computers through their support. I would say ~~essentially~~ ^{basically} what they did was to develop hardware, ^{develop} languages with which to program, and they stopped there. But what they should ~~have~~ done was to develop good procedures and standards for using these languages. They never did that. They just would give you a language and ~~say~~ go ahead, do what you want with it. Because of this, programming has become, to a very large degree, an art. Each programmer has his own

(more)

concept of how to organize a program, how to debug, how to document, and it's the manufacturer who never created the discipline that was needed. Now we're essentially a runaway industry, where companies or organizations have little muscle with which to do anything.

CHRISTIAN: Will "contract programming" remaining essentially a "cottage" industry?

WELKE: This term comes ~~from~~ out of the textile ~~industry~~ or dressmaking industry, where you used to take the product into the house and let the women sew and knit and turn out three dresses at the end of the day.

CHRISTIAN: Will it be the customer or the package industry that emerges here?

WELKE: What it gets down to is, are we going to wind up with a dozen big software firms that are national in scope in every respect? Are we going to have a lot of small problems, or possibly large problems, but limited to their geographic area ~~or~~ or a particular segment of an industry? Is it going to be small units or large?

DREYER: I'll start the dispute rolling. I'll just mention the survey that ADAPSO is issuing this month based on information from the time sharing, batch processing ~~part~~ and the software segment of the industry. Our long range projection for the software ~~segment~~ segment, as it is for the batch processing part of the industry; is that there will be approximately 100 major firms, down from 800 at the present time, controlling the enterprises that we're involved in. We estimate for 1970 for example, that the software

(more)

segment of the industry will do \$800 million. The batch processing part, a billion and three-quarters, time sharing ~~and~~ around \$300 million. By 1975, the total computer services industry will do about \$5 million with many less ~~com~~panies and many more branches. I'll throw that on the table as a starting point.

ROWELL: If we look at the transition of the hardware industry and what happened to the requirement for software; we'll see a movement toward the ^uproliferation of both large and small special-purpose machines. In going to the big machines, we have created operating systems which are, in themselves, general-purpose software. A major company or two is delivering that ~~the~~ operating system, thus reducing the need for a great part of the major programming that is being carried out. We are creating large general-purpose software, in the main, to take care of the vast number of firms that will be able to do ^{The type} ~~such kind~~ of software that we're talking about. Then the local "cottage-type" industry will probably continue.

DREYER: One other point. We also anticipate in our survey, by 1973, a shortage of about 250,000 programming and systems people. It would be rather difficult at this point to say where the gap is going to ~~be~~ be filled.

ROWELL: Are ~~you~~ you indicating that there's a small number of large firms, but also the continuing existence of the "cottage" industry?

DREYER: Yes, And the same applies to the data processing part of the business. There will be the utilities, the time sharing elements, and also the local market.

AHLERS: I'm beginning to discover that I may be the only user here, not the
(more)

(Handwritten signature)

only abuser, I'm sure. The emphasis will be completely on applications, and this is not a function of size. In our situation, in banking, where we're dealing with everything from mergers to investment banking it's the three or four guys who really understand the application who are critical; and the 400 or 500 programmers who are perhaps not critical. It's not a question of numbers, it's a question of knowledge about the application. Along with this goes technical knowledge. I'm not saying I think there's some sort of minimal entry cost in an economic sense that you can get in this business. I think we will have the specialist in this field for some time.

GOETZ: There are many kinds of proprietary software to build and generally what you need is the expertise in specific areas. It is very difficult to get one company that would be experienced in all the different areas. I can see 50, 100, or 200 different firms doing certain aspects, building certain ^k kinds of software to meet different needs. In that way companies will compete very successfully with IBM, in terms of proprietary software. If you look at the various kinds of software that should be built you might get anywhere from 100 to 500 different kinds of packages, as many as 1,000 major kinds of software that should be built. I think it is too much to expect one or several companies to be able to build it and do a good job.

SCHOENBERG: I think the question here implies that contract programming will remain itself. I personally believe it will but I know there are many people in our industry who believe that contract programming itself is not a very viable entity. Almost every company who does contract programming is presently attempting to provide a broader service. I'm not so sure that that is because they believe that contract programming itself isn't viable, or rather that in order to more readily market your services in a variety of areas it turns out easier to provide a broader

(more)

range of services. When we talk about the question of remaining cottage size, I think everyone in the industry always feels that they are going to get larger. There are some very big competitors who make statements about how they don't understand how any new companies can start in business. It's remarkable how they continually manage to start and compete effectively. ^{Something is left out.} There are a lot of rationales for this. The most obvious one to me is the typical size of client. There is something peculiar about a contractual arrangement between a company that does \$1 million or \$2 million a year and a company that's doing \$1 billion a year. There is actually almost no normal recourse for the large company, which is naturally a very bad situation.

CHRISTIAN: From my point of view, as an editor, I've noticed much less information input from software companies since unbundling was announced.

SCHOENBERG: We've had two specific experiences that I would relate to unbundling. We had a very large contract that we presumed we were going to get, but eventually lost to IBM. It was quite a bit larger than anything that we've ever had before. I would say that the first effect we had was a very negative one, discovering that IBM can be a very potent competitor. However, we've had a much more favorable kind of problem. We were dealing with a relatively small company: not even a company, a non-profit organization. I'm sure three months ago, if we had gone in and asked for our typical rates, we would have been laughed at. They would have said, "Go away, we can't pay that kind of money." We asked for more than our normal rate and we got immediately accepted. When I tried to find out why, it came out rather easily. IBM had until recently given them someone on a systems engineering basis and was of course ^gchanging significantly more. So I think as in the past they tend to create an acceptability for the market, which is making things a lot easier.

(more)

GRINKER: Cottage industries are run by cottage businessmen. Your observation, Larry, is that you ran into a non-cottage businessman, you ran into IBM. And they are not cottage businessmen. They understand capital-- you notice that they usually don't allow a non-profitable cottage to exist very long. They are capable of pulling the chain on any number of businesses, or products, or functions within their company, by measuring it on a profitability basis. If they are in any non-profit segments of the business it is by choice. They fully understand that it is non-profit. But the cottage nature of the software business right now is such that it is not really capital-requiring. You don't really need a lot of capital to build a small software company. And I'm not referring to any major capital-- more than most people think, but certainly nothing consistent with the capital requirements of the Telephone Company, or U.S. Steel, or heavy industry. It's certainly not free. One of the big problems we find in our trade-consulting practice with companies in the business is, when you give them a hunk of money they really don't know what to do with it, they don't know how to spend it well.

GOETZ: As far as ADR is concerned, at this point ~~unbundling~~^{by} bundling has not been a factor in our business. Essentially we're trying to market and develop systems software. To date, virtually all systems software is still given away free, so it has not affected our business. I agree with Larry Schoenberg that contract programming has a better environment since IBM's price structure is higher than that which the average small service firm charges. But other than that it has had no effect on our business, good or bad.

CHRISTIAN: The effect of unbundling was to improve the price ratio for the software companies so they could charge more for their services.

GRINKER: That was not the intention of unbundling, incidentally. Unbundling

(more)

was done by IBM. The decision was a completely internal one to allow the IBM Company to pursue separate courses of action and to announce in particular its System/3.

DREYER: I'd have to disagree with that statement, because ADAPSO was very much involved with unbundling. We put out a position paper six months before the unbundling was announced. We were down at the Justice Department at which time we were told, off the record, that by July of this year action was going to be taken by the Justice Department if unbundling hadn't occurred. Let's call it a "voluntary" unbundling.

There was an interesting reaction by the members of ADAPSO. The three percent reduction that occurred on the surface appeared to be a reduction, but then look at the systems engineers involved, the education, and other things. However we found that they didn't use the SE's very much, they didn't use education facilities very much. What happened with the three percent was that those companies with in-house possibilities didn't think much of that reduction and therefore they didn't have any installations made and therefore kept our market available. A second thing was that many of our members felt a new profit center had developed. IBM was charging about \$28 an hour for software development. All of a sudden our people who had been giving away software or charging a very minimal price were now starting to ask \$14 to \$18 an hour. It was a plus factor.

FRANK: A number of interesting points have been raised in regard to cottage industry and they're all related in a way. Going back to the first point which is, "Will the cottage industry remain?" I think yes. Look at some of the reasons why contract programming services are used today. Many of those will still remain in effect 4 - 6 years from now.

(more)

How will unbundling affect this? Unbundling, as Larry has indicated, has already begun to create an umbrella. There are some other interesting considerations and facts that at least we have begun to feel, in dealing with users. One is, unbundling has-- while it may create a profit umbrella-- I'm not sure that it will be a continuing profit umbrella, in the sense that users are beginning to evaluate IBM on a man-for-man basis with a much more critical eye. We, for example, recently got a modest-sized contract where IBM we found out later, had been called in first. Apparently, because it was an IBM program that was being modified. The idea was that, even though at IBM, the same people who originally wrote the IBM program would not write this one, they would at least have access to the people who did. The thinking there was that basically IBM would be able to do the job better, more quickly, and perhaps even more efficiently than anyone else. Once IBM got back with their proposal the user realized, at least to his satisfaction, at that point, that the proposal was not responsive and he went outside.

ROWELL: A point that I don't think I've heard anybody make in terms of unbundling, but one that we began to see, is the fact that most managers of data processing facilities around the country have been recognizing the fact that hardware costs money and programmers cost money on a salary basis. It's pulled together by unbundling. Managers are beginning to realize that the development of programs also costs money. I've heard many managers say, "Well, I've already got programmers on my staff, so I'll just go ahead and develop that software myself." A recognition of the fact that a program or a package of programs has itself a continuous dollar flow is probably going to raise the demand for contract programming.

(more)

CHRISTIAN: When, if ever, will programming be done by non-technicians, by laymen, or the public? In other words, when are people going to stop communicating with machines and machines start communicating with people?

FRANK: They will obviously, because as languages become higher level and easier for people to use in a variety of situations, there will be more motivation and opportunity for people to work with non in-depth technical languages. Education, starting in the high schools and going to the colleges, will give people the knowledge and the tools that they need to interface on a daily basis with a system.

AHLERS: Actually we've already done this in the banking industry, in a large sense thanks to the ABA, by trying to identify a particular financial area. We've had vice presidents take a hard look at what it means to lease computers, to evaluate proposals, and merger deals in the period of hours that before would be literally impossible. One thing that really bothers me is will programming be done by non-technicians? Personally, I don't see much of a future for programming, "period." It seems to me that programming is a very structured sort of thing that computers do rather well. What computers don't do well is structure the problem. I don't see very much of a future for natural languages either, where you sit down at the computer and you say, "Please, let me work with the following kind of a financial problem," and the computer replies, "Yes, what is your problem?"

GRINKER: What you have seen is a number of efforts die on the vine because middle-managers tend to look at a computer and a terminal as a toy. They really become bored after a while. The time sharing terminal has become sort of the office toy, with about the same life-cycle. What may happen is that over several generations you're going to find that non-technicians won't use computers, but that managers will become technicians because of the technology of our society. I don't think the devices will become simpler so that any idiot can use them, I just think that the simple idiots will become a little bit smarter.

(more)

McWILLIAMS: All programming eventually will be done only by technicians. Maybe at some point in time, there won't be any programming, and therefore no technicians. You're not talking about, when you say you're developing some higher-level language or some means of enabling bank executives-- management-- to communicate with the computer, you're not talking about programming, as we know it. I'm very interested to hear that the ABA is doing all these things, but it's not very prevalent around the industry. Let's go ahead a moment. Another thing, with respect to the using public, is it will evolve to the point where managers come to recognize that they must be able to communicate immediately with the computer. They must, to provide a viable business. Whether they become technicians as a result of doing that I don't know. But the biggest challenge management faces in the '70's so to speak, is not having the DP operation sit out in one corner of the building and muck around with an MIS system and come up with a fancier way of production scheduling. They have to realize that they have to get down with that thing and use it to manage their business or that business won't survive.

GOETZ: Without trying to complicate the question, I think the problem of defining programming, as we know it, will always be done by programmers. Programming done today, and perhaps the example that the gentleman from Bankers Trust used was a good example, is not really programming, it's feeding data in a simplified form into a program that was written by programmers. In summary, perhaps the amount of programming as we know it will be reduced proportionately. In other words, there's a great deal of overlap, a great deal of people writing the same kinds of programs. I think programming will be done, perhaps not as much coding as we know it, but certainly programming will be done by professional people.

(more)

CHRISTIAN: What will be the impact of the large, Fortune 500-type companies entering the software industry?

DREYER: ADAPSO, as I'm sure you all know, has some very strong feelings on this subject. We divide our feelings in two ways. First, there are the free enterprise companies. The others are those that have types of government monopolies. Free enterprise companies, we feel, can belong to any business, under the business rules, and diversify their assets as they see fit. Government-sponsored granted organizations, such as common communication carriers and the banking industry have certain rules and regulations that ^assist them. So our attitude is if a company is in the free enterprise category, he's certainly welcome to fight the competition in whatever aspect of the industry he cares to join in.

ROWELL: Let me just give a case study without any commitment one way or the other. COSMIC is a non-profit corporate affiliate of the University of Georgia that was started a little over three years ago. The person that started it had a good business background. He wound up with ulcers in six months and left. The principle of the facility was, with its ^{the} academic cloak, to make available a transfer facility for tax-paid developments in the area of computer software: starting on a small-scale approach with NASA, taking the software that NASA developed, rebuilding it, evaluating it, finishing off the documentation, generalizing hopefully, and selling it to support the function. It moved into an affiliation with the Department of Defense and the Atomic Energy Commission. The government is not a Fortune 500 company, but it certainly is one of the big ones.

At this point there is a question as to whether it's terribly successful, this function of transferring software. The price function is the reason it's not terribly successful at this point. We're only running at about a million dollars a year. We've got government controls on the prices we can ^Charge for the material. Let's say that the government is
(more)

developing software for its own use. We're taking that software and redistributing it at a price to people. We've come up with some very useful packages, some very high demand packages, some packages that are in fact in competition with some packages that some of you gentlemen in the room handle. If the government can do it, and if programmed products are being developed for government use that can be resold, then it can be done for the Fortune 500 companies, too.

CHRISTIAN: How do you think the user feels about dealing with the Fortune 500 company buying his package, as opposed to one from an independent vendor?

SCHOENBERG: The reason why we aren't terribly concerned about most of the Fortune 500 companies competing is that our business is still a very marketing-oriented business. It isn't a question really of what a customer would do if he had that alternative of buying from a Fortune 500 company, but whether he is going to know about it. In fact, most companies that have formed this kind of group have not done very well. Those that have done well have done so on the stock market more than any other aspect.

I think that if it is going to be a question of packages, it strikes me they wouldn't have ^{ch} ~~must~~ of an advantage, since a package is presumed to be examined by itself. It would be more significant in terms of contract programming, where there is a question of reliability and stability, and so forth. Our experience to date, although it may be a little bit short, has been that these companies do not have ^t ~~than~~ entrepreneurial spirit, for rather obvious reasons. They simply are not very effective competitors.

McWILLIAMS: I don't discern any trend among the Fortune 500 companies to take this step, although I believe there are many significant instances of it. I think there are some trends within certain industry categories toward the software industry: finance, and probably in the publishing

(more)

industry, and for very good reasons in both areas. The short-range impact on the software industry will be the following: that those who are really committed to being in the information-processing industry, or software industry, very likely will take advantage of the fallout that is occurring and will continue to occur, because of underfinanced good ideas but not enough marketing. All the reasons that companies will be on the verge of going under in the next couple of years. We'll see them disappear into some larger company. You're seeing some disappear into the insurance banking and publishing industry.) Although the software computer industry is at this point a non-capital goods industry, there is an awareness of and a need for, resources available to them on a long range capital investment program. When I talk about long range I mean, as an example in the publishing industry. They invest in forests that pay off after 50 - 60 years. In the oil industry they have 100-year capital investment programs. People that have an attitude and an understanding and the guts to go into something with the promise held by our industry and stick to it, will be very forceful competitors.

GRINKER: The Fortune 500 company, the large capital resource type company coming into the software business, is going to create a fantastic business opportunity for those people already in this line. I feel quite fortunate. We have a software marketing company that we've been quite successful with. At this point, because of the Fortune 500 companies coming in, and the undercapitalized small software producer. I have my choice of some 300 products right now that I can bring to the market on my own terms and strike my own deal. The diversity of choice is substantially open to me. What is the impact going to be? I think it is going to force the independent software business into addressing itself to the marketing and distribution questions, rather than the product

(more)

creation questions. The software business being one that sprung out of the technician. If you will, he has addressed himself over the '60's to the question of creation. Very few companies have addressed themselves to the question of distribution. With the products becoming tremendously prevalent, you no longer have to address yourself principally to creation, but far more importantly, address yourself to distribution. The companies with the distribution capability will have far more freedom of choice to exercise their distribution organizations and to get a much better return on their organizational investments. They won't have to wait until the boys in the back room discover another "gee-whiz!" package. They will just have that much merchandise available. Fortune 500 companies don't really want to distribute, maintain or sustain the business relationship that selling services and delivering services required.

GOETZ: The Fortune 500 companies will be a force only in that ^{the} they will acquire companies and perhaps have many independent subsidiaries that might provide in-house services and also outside services. I don't think they'll be successful if they try to internally develop software companies through some of their key people, and essentially try to develop new companies. I think if we look back over the years in terms of many large Fortune 500 companies that entered the hardware field, it's still a very competitive area. If you look at Ford acquiring Philco, there are still investments and management required. I have my doubts about GE. They're in and out of the field. I don't think it's size alone. I think it's a dedication and a growth potential of the key people within a company that will make the software companies grow.

#