[THIS IS AN INTERVIEW WITH DAVID STONE, VICE PRESIDENT OF THE NEW SOFTWARE GROUP IN SPITBROOK ROAD, CKO FACILITY, SEPTEMBER 16TH, 9 O'CLOCK IN THE MORNING].

BL: I'd like to start when you first fell into Digital. I heard you had designed operating systems.

DS: I went to Harvard and I studied mathematics and I got involved in using computers there. It was back when we were using punchcards and signing up at 2 o'clock in the morning when the computers were available. When I went to select my career I had a choice of a technical career with Bell Labs or with the bank. And I went the technical route. The project they put me on was Multix. That was back when Bell Labs and GE and MIT were collaborating. And they were trying to design the ultimate time-sharing system I think you probably could say, the system of the future. So, I was lucky enough to do my original work as a programmer in a research environment talking about concepts, some of which are so advanced that they still haven't been implemented. I came out of there, I got a degree from the University of Pennsylvania at the same time with Bell Labs. I came out of there with a fantastic technical background. I took that to various places. I worked at Raytheon just before DEC and there I was building guided missile _________ systems. The trouble with that is that you get into all these silly things like body counts and so on, that wasn't much fun. So I came to DEC, an ad in the New York Times which Larry Portner put in which he called - "Portner's Complaint', where was he going to get all these programmers he needed. And that just happened to hit me when I was interested. So I came in 1970 and I was running the DEC-10 Operating System Group and I was able to translate a number of research learnings from Multix into the design of that system almost instantly. And, for example, I have a patent, over there on the wall, the middle one there, which is on the P-1145(?) Memory Management Unit. I'm not a hardware person but I was doing was explaining to the hardware people what an operating system like Multix would want to have in its hardware.

BL: That was Bob Stewart I think.

DS: The patent is Cutler and Gray and myself. I think Gray was the person who was actually doing the hardware. Bob Stewart certainly was involved but I think it was Gray who did it. In any case, that was how I got into the hardware side of the angle just because I was trying to make sure that the operating system would have the tools. Other people took the Multix learning outside of the company and then they used that machine because they could see that was what it was for. So it was kind of interesting, the way the Multix alumni are still widely spread and quite influential.

BL: That must be circa, late 1970s?

DS: No it was early, I moved to Europe in '74. So I think the patent itself was probably applied for in '72. The patent was received on December 31st, 1974 so that probably means we submitted it in '72. So I started off working on the Operating System Group that DEC had, very quickly I took responsibility for the whole of DEC ____ software ______ products as well as the operating system. And almost instantly after that I was promoted to take responsibility for all of the software engineering in the company, all the operating systems, all the hardware, all the software, all hardware units. And at the time we had PDP-8, PDP-10, PDP-11, PDP-15. And that was just before we started to merge everything on the PDP-11 line. It started out looking like four different businesses with four completely different sets of platforms. We had an operating system for every basic compiler which meant there were twelve operating systems. It was a very different environment. Then in '74 I went to Europe to run software services and engineering in Europe and in '72 I had started up the first engineering activity in Europe which was in Reading. So the Reading facility I started up and then I went over to Europe and managed that as well as some other things. It was a very interesting time.

BL: You started in the beginnings of DEC-10, just a barely running operating system from the 6 I believe.

DS: Yeah, well actually when I first came in they were just about to release what was called the Level 5 File System. So, in 1970, they already had a working system that worked pretty well. They were trying to introduce an operating system which had a file system with what's called, - fine grain locking. They wanted to have many people accessing things in the file system and not getting in each other's way and not being blocked out because one person was accessing it. And the research on that topic was called - "Deadly Embrace" - the theory was if you tried to access resources and the other person was trying to access resources, you could take A and he could take B and you could fight each other over who would get the other one and, and completely stop progress. And that was so called - "Deadly Embrace". And they had not resolved that from a research perspective and I knew that because I had been working on it. So I came in and basically put a lock in front and a lock in back of the file system and it worked. And it took them four years to actually figure out how to make it work. So they were trying to do something that was very clever but we just didn't know how to do it at the time.

BL: What was the major difference between the development of the 10 operating systems and what would become the multiple operating systems for the 11?

DS: Well, the 10 was competing in an environment which was much more demanding, in a sense, there were more things; that were wanted: it had to do batch, it had to do real time, it had to do time-sharing. It was particularly good at time-sharing, that was its main function. And we kept expanding it outward to cover all those things. So it was more like a precursor to the MS than it was to the existing 11 systems. So it was more robust. It was used in more general purpose environments, let's say. As I say, it was primarily a time-sharing class of system. The PDP-11 systems RSSX was basically a real time system. It covered more territory, it was more complicated, it had more code in it and it tried to do more things for more people. From a design perspective, you can design operating systems in one of three ways; you can either focus on real time which means the highest priority gets all the resources and that's the end of the story, or you can focus on the batch which was the classic way it had been done and that says you optimize the use of the computing hardware at the expense of priority for the users for the jobs. And then there is the time-sharing version where you optimize on the perception by the user, so you want to give him a little fair slice every now and again. And the DEC-10 was first and foremost oriented on the user model, on the time-sharing model and then branched out to the batch in real time. Whereas the 11 and also MS were based primarily on the real time model and then they expanded to cover the other possibilities.

BL: Very interesting, I get the feeling that in terms of the architectures of the two systems in this case, software was very much an after thought. It seems as if the 10 and 11 there was an elegant hardware texture put in place then it was thrown over to the land of software, to you to see what can you do with this?

DS: To a certain extent that's true. Now, look at VMS, there are a lot of instructions in the VAX that were specifically put in for the software people. The so-called complex instructions of computing is complex because you put in a lot of things the software _____ ____. The MS was the opposite. There, they really concerned themselves with having software ______. The DEC-10 there was some good cooperation but by and large I think you're right. In the 10 and the 11 it was the hardware first and then thrown over to the software people. I'll give you an example, - at one point the question was about disks. You have disks and these disks can either go twice as fast, spinning twice as fast or you can stick twice as much stuff on them. In both cases the amount of stuff that flows by, the head is the same and in one case the physical disks would be faster. And the question was, - which of those two things should be done? And interesting enough, from the hardware perspective they can say, - well, what we really wanted to do was to make it go faster. And I said, - well, I'll tell you, from a software perspective that's the wrong choice. A software person will tell you to put twice as much data on it and then I can simulate it going faster by spreading the data twice on the disk, once on each half. So from a software point of view, you have more flexibility just by stuffing it on the disk. The hardware people actually made it go faster 'cause they thought that was the thing that people wanted to buy. There was a case where it wasn't perfectly integrated.

BL: How well did the first version of the operating system, te 10 run? In fact was it called TOPS at that point?DS: No TOPS was later on. The TOPS-10 System and then TOPS-20. It was just an ______. At a certain point we actually got some technology in from BDD&N(?) which I went and negotiated with people at BDD&N, some of whom have joined us since.

BL: Networking ____ _____?

DS: Not in the operating system side. He used to work on the 10 for a long, long time, I can see his face, I can't remember his name at the moment. I'll recall it for you. There was a pretty early evolution and it was typically in terms of level ____, [UNINTELLIGIBLE]. It became TOPS at the marketing, the 10 Operating. Then when the 20 System came in, that was another minor evolution. There were a lot of hardware pieces that came along: KA-10, KI-10, all the complexities of the hardware were exploited in the operating system. It was a good evolutionary trend of taking the latest technology and making it available in the operating system and therefore to the users. It got very expensive. The fundamental thing that cause the DEC-10 line to fail is that there was a big fight in the beginning about whether we should do 32 bits or 36 bits and we went to VAX and that occurred around the time I went to Europe so I played in the beginning but I didn't play in the end of that discussion. The argument was, that we could take the DEC-10 and it's stuff and use it because it worked and it was very complex, it would be hard to duplicate. On the other hand, the world seemed to be going for this multiple of 8 bites, which would get you to 32. In the end they went to 32. And what was going to save the 10 was they were going to have a 15 nanosecond processor cycle ____ which was beyond the state of the art at the time and proved to be impossible for them to generate.

That was the so-called Jupiter hardware system. We tried to do that for a long time and then finally bit the bullet and said we didn't know how to do it. Told the customers and they were duly unhappy and we spent a lot of time bleeding in front of the customers, on their behalf and then eventually we were able to move most of that over to VMS which is where we focused.

BL: Speaking of customers, how close did you feel you were to customers understanding their needs? What was the role of providing anything more than operating systems in that timeframe? There's a little in the middle where there are some compilers.

DS: There were compilers and there were file systems. The human interface was hopelessly primitive by today's standards. One of the interesting cases that happened was when we started to combine the command languages for the 8 and the 11, the 15 and the 10. And the 15 eventually went away. We basically took the brightest people from each of the groups: a Richie Larry and a Tom Hastings and a Dave Cutler and so on; locked them in a room and said you come out when you have a common command language and that was how DCL was born, Digital Command Language. So we did have a number of principals of the middle ware but the middle ware was all stuck into the operating system. The reason it's harder to see is that since everything ran on one note, on one box, the middle ware and the operating system were glommed together. Today, with the client-server architectures you see increasingly the middle ware is pulled out ___ _____ ___. There was more middle ware than would be apparent but we never unbundled it because there was nothing common between the system. But, yes, mainly you were delivering file storage and retrieval and the fundamental complexity masking software for the hardware, worrying about all the hardware instructions and interrupts and IO and whatever that would mean. Then you started adding things like batch systems so that you could let people pretend they were running an IBM type environment on top of our equipment. But the big advantage, the things that customers wanted, was the development environment being able to do things faster so the time-sharing capability would allow people to develop programs much, much more quickly. I programmed for example on an IBM machine before I came to DEC. And what you did was batch and you had this shelf about a yard wide of manuals which were explaining their job control language similar to our DCL. And that was the hardest part. You would be turning these batch jobs around and you'd stay all night and you'd put a job in and you'd get coffee and wait. And the magic of this time-sharing system was, you didn't have to wait, at least not very long, for the job didn't take too long to compile. That was where people wanted it. It was a wonderfully close community. The DEC-10 group at DECUS basically was all of the ______ research of Ken. So there was a very, very close feedback. We had groups from the customers telling us what they wanted. And if they all agreed, I always told them, if you all agree on something we'll do it. How could we not? We had perfect and instantaneous feed-back from the majority of the users. And that was something we've lost as we became much larger and covered so many markets.

BL: That feedback was to provide more and more of the utilities middle ware.DS: Right, they would say, - here's the problem we're trying to solve. Here are the things we've been having to do. We'd like you to do this, add this feature and that feature. It was very clear that the people who ran those machines were telling us what they needed to get their job done. And that's the purest kind of market research you could have, they're willing to actually tell you. So that was fun.

BL: What did you find in Europe?DS: Well when I went to Europe I was looking from the software services side. There, there were a hundred people. When I started in DEC in 1970 there were 100 people in all of software engineering. When I left in '74 there were about 300. When I went to Europe there were a hundred people in software services in Europe and by the time I left, I stopped managing that in '84 which is about ten years, there were three thousand. So there was pretty rapid growth there. In Europe we did a very small amount of engineering so that really wasn't very significant. What I was really doing was dealing with large projects but by today's standards they were not very large but dealing with sales of things and supporting them. We had just begun to try to make software service into a business. We were just beginning to charge for, software maintenance, for example. I remember the year I was there I think we just made, in all of Europe, a hundred thousand dollars on service contracts for software maintenance. Today, we are probably making five hundred million dollars on such contracts. So it's moved up just a trifle.

BL: Which is a fascinating change from the idea of software being free.

DS: Oh you bet. We pioneered, people forget sometimes, we talk about having to change the way we license software and people say, well nobody will accept it. We were the first people to charge for operating systems.

BL: In what year? What operating system? How did that mind set change?DS: Well before my time we'd started charging for hardware service. Hardware service in the beginning of time was _____. We started charging for hardware service, made it very profitable. It's a major profit generator for the company today. Then we started charging for operating systems and that must have been somewhere either in the very late sixties or the very early seventies. The DEC-10 operating system was only running on the DEC-10. The question was, - would people want to buy the hardware without the operating systems? The answer was yes. Sometimes they wanted to write their own. People used to write their own operating systems because they were simpler. In those days I really can't remember when it came exactly but it must have been very early seventies or the late sixties that it happened. Then, all the other software pieces started to break out and get charged for it. We were the first people who charged for software maintenance and I believe that was in '72. Each time people said, - the customers will never accept it. And each time as soon as we did it everybody else did it too. So we have been leaders in changing the world's perception of how you can charge for these things. But as we got into the commercial marketplace, we stopped being a leader of how things were perceived. In the technical marketplace we could do that. In the marketplace that we're in now, we cannot do that, we do not have that market position to allow us to do that.

BL: Yet, I see the roots of our strength in what's called, system integration, tools or _____ support that, being a very big part of the competitive advantage especially in Europe when you went to start working in projects(?). Software services I guess could be systems integration.

DS: That was the very beginning, software services was doing pre-sales work, was doing consulting work and it was doing project work. At that time the biggest thing was clearly the pre-sales and software maintenance side, those were the two ____. We began taking advantage of our strength there and doing systems integration more and more. Now I think we're the fourth or fifth largest systems integrator in the world, according to Data(?) _______.

BL: What was your focus in Europe aside from building more and more software service(?) capability?DS: That was the primary thing. I did engineering as well for much of the time. It was a small activity but I managed the growth of that. In 1984, during that time various other responsibilities, I rand the MIS function for a year in addition to the service function. And I ran the marketing for office and telecommunications in 1984 - right next to my patent over there on the right, the market year of the year 1984 and that was fun. That was selling All-in-One VAC(?) before we had it and then in '86 I started doing metaframe activities which was strategic planning and collaboration work. And I worked on a number of major exercises for that creating the insight into where things would go, developing for the initial concepts of the computer utility that we talk about today and trying to map how we would do business, got a second patent on the process that I created for displaying how you would set up a strategic collaboration with another company. The patent is really quite funny because I went to my patent lawyers from a software services perspective and said, - we need to learn how to patent a process, the way that we do things. And they said, - you can't do that. That's not the way patents work. So we said, - how do we protect it? And they talked to me about trade secrets and copyrights and so on. And eventually I said, - there must a way. Here's this process I've developed, it's really good. How can we do it? So in the end they found a way. The way we actually got the patent is the following, - you take a company's capabilities, you slice up the universe of all the little things you would like to have in an information technology company and then you map, against those little box of capabilities, a big plastic overlay of a given company and you put a color, let's say blue for DEC and in each box you have as much blue as is appropriate to their capability there. And you take another company, let's say yellow and you put another overlay. Now you can see how good it is. You put the two of them together and you green on the overlap. And the patent is actually the fact that you can see this green where the overlap is. Just recently the Zip-Lock baggy people came out with this same patent with their closing, they say the yellow and the blue, you close it, it's green. So I thought about suing them but I decided it wasn't worth it. You'd never know this from reading the patent. Let me read the patent, it says - "A method of apparatus is disclosed for producing comparative data in a visually assimilable form. The method compromises the body of substrata which is divided into visually distinct zones" ... blah, blah, blah. This goes on and on. If you put blue and yellow together then you get green.

BL: It's brilliant.

DS: Funny stuff.

BL: Did you say for market mapping?DS: You had a market map which you then overlaid with the capabilities of a given company. If you wanted to compare two companies, say, - would these two companies be able to work together well or would there be too much conflict? And so the green told you there was too much conflict is there was too much green.

BL: '84, Skippy Walters tells a wonderful tale of walking off the plane one morning in Spain and being dragged to the meeting by you, big annual meeting. He asked you, - what should I do? What should I do? And you just said, - just emphasize the consonants. And then he said, you threw him up in front of this group of all the sales people and he just went through his regular rap. All of a sudden everyone started getting real excited. He didn't know why, just talking the usual All-in-One stuff, it was never very well received. All of a sudden, somebody asked him a question, he said it's easy to translate this into _____ languages obviously. Oh yeah, I guess that would be interesting. He hadn't really thought about it too much. But then he watched you turn this crowd of people over the next day or so into a thriving mass of people. People running out, coming back and created the most exciting marketing event that he had ever seen in his life.

DS: Oh that was great fun. That was in Marbilla(?), I'm sorry where was it? That was on one of the little island just.

BL: Majorca.

DS: Majorca, yes. That was a wonderful meeting. We had all the software services managers across Europe. It was the annual meeting of the group aligning the way for the future. At the time we were having one of our periodic crisis. So we were looking for ways to energize people and to go out and sell more. So we created the, - "How To Sell A VAX In 90 Days" campaign around All-in-One. And we got everybody to sign up. As Skip says, we generated the enthusiasm, for this was something we could do for the company and that we were uniquely positioned to do because it was software services and software services had built the original All-in-One at DuPont. And Skip came and talked to people about what it was. We got all enthusiastic and we had workshops and we designed T-shirts and this was back before the T-shirts were available on every street corner. So we went off and found somebody and got them and we designed logos and we set up special sales update articles. It was really exciting because it was a business opportunity that people saw and everybody went away very energized, very energized. Skip was a big help. That was one of the high points of such meetings in my life too. I've run a lot of those meetings and that was one where the people got more involved. And we were very successful in meeting our objectives around selling All-in-Ones that year, using that enthusiasm and we got the salesforces involved and it was great fun. We had all kinds of events in the evenings re-emphasizing these tings. We had stories told and parables created and then every now and again they had the flamenco dancers who would provide a little break from the intellectual activity. That was a wonderful time. It's an example of what you can really do with a group that gets aligned around a mission, something that I think today we're lacking to a certain extent, that unifying theme that allows us to move ahead and really deliver our own part of something that we can see the whole of.

BL: Tell me about other examples of the company or groups in the company being aligned around a mission that you're all part of.

DS: The other piece that's really significant is, all of Europe, where __Carlo Felotti(?) took over Europe in '83 I thin it was. One of the things that he asked me to do was to start this metaframe(?) activity. And along those lines we created a mission and a set of objectives for all of Europe. In fact I have the latest version. Here's the example of the latest one. We created that with the whole management team, it took 20 of us about four days and I ran the process for that. We walked out of there with all the ______ agreed. It's done, no more staff work, the whole thing was done. I was the only one on our team who could type, being a programmer, overnight

I would take all the stuff and type it up and distribute. So we had in the end a complete set of mission objectives. And I believe and __Carlo believes that that unifying vision of where we going and how we were going to get there, made the big difference in why Europe grew from maybe 30 percent of our revenues and now it's probably 45 or almost 50 percent and how we got the groups to align because we had in there all kinds of clear instructions, why do you have this function if this function can do this work. The interfunctional work was clearly specified. How you'd work together, team work, communications, performances, all were specified. I think that was probably the biggest, most significant example of the major portion of the company aligning around the common vision.

BL: Perhaps there is something uniquely European in that there are so many different companies but differences are respected. Everyone is very happy to work together. Define how you're going to work together. [INAUDIBLE].

DS: I think the way I think about that is, in Europe and in fact most of the world, it's perfectly clear that unless things are clearly defined you can't cooperate. In the U.S. there is, what I would call, sloppy thinking. You just make a phone call and everybody meets together in a room. And somehow or other things will happen. It doesn't work that way in fact in the U.S. either but in Europe because of different languages and cultures, you know it can't work. So the advantage you have working in Europe, since you know it can't work unless it's formally defined, you do more work on the definition and you are more careful with the language since you know people are more likely to misinterpret. I think there's a lot to learn that way. One of the things I learned in Europe was that because of the countries, the requirement interface with countries, you had stronger people further down in the organization because you had a higher requirement on them, not only to do the Digital thing but represent Digital to the outside, set prices and so on. Therefore you had a stronger organization even though it was smaller than the U.S., where many more things were centralized because you could do that with one country. I think that's the major difference in the U.S. ______ versus the international view of how things should be.

BL: From the framework work in '86 what have your driving forth events(?)?

DS: For me personally? Well since '86 I spent a lot of time on the mediframe(?) activity. I spent a lot of time on marketing. I once again did marketing for telecommunications in '89. That's my other award over there, the marketeer of the year '89. While I was doing engineering, I did international engineering starting in '84. There the problem was, we had never thought about designing our products so they could be translated. For example, you took representation of characters, on a terminal like the BT-100 you had the ability to run 7 bit characters. And that was fine for handling all the U.S. characters, all the ASCII or ____ ____. However when you went to Europe and you had all of these various other dia(?) critical marks like circumflex and ...

[END OF SIDE 1].

BL: It required more, there was no international character set standard?

DS: There was no standard at the time and people were just talking about the standard. What they did was they would replace certain characters in the standard American set for a given country. You could do that. So you had enough for any given country but when you wanted to do all the countries, you didn't have enough 'cause they were using the same characters. So, a seven bit character set wouldn't work in a multi-national environment. You had to decide which country you were in. So then we said, we'd like to have a character set that had all of these characters and you wouldn't have to worry about which country you were in. We got tangled up because we had, for example, the DECmate word processing, obviously the characters are very important. That was a 7 bit machine in terms of character set. Then you had the new terminals, the VT-200 series, which we convinced them to use 8 bit character set. But they didn't do the 7 bit character set. So now you couldn't have both DECmates and VT-220s on the same system in Europe because they expected different kinds of character sets. We had to invent software that would figure out which it was. So it was very complicated. By today's standard, we didn't think about it all. Now we got pretty good rules for it, we sell books on how to produce international software. International engineering, both hardware and software, was something I was looking at. The second thing I was doing in engineering was worrying about the needs for installing engineering activities in the countries due to marketing constraints from governments. So the government would say, - if you don't put engineering work in my country then I won't buy from you. And different countries had different degrees of pressure there. Most of the pressure was in those countries where they had a local computer supplier. So France, Germany, UK, Spain were the four really big ones where the pressure was. One of my tasks was to get engineering to lure engineering into Europe. We were quite successful. We moved up to the point where engineering in Europe was sort of negligible to about 2 percent of European NOR(?). By comparison, in the U.S. it was like 20 percent of U.S.

NOR(?). So that was one way to measure and we were pretty successful at that for quite a long time. The international engineering part, the engineering presence part and then of course providing the environment so that all the engineering groups could use _____ products. Between marketing activities from time to time, the engineering activities I've just described and then the strategic planning activities that took most of my energy. I came back a year ago, in 1990 in June.
BL: It seems a lot of this is providing specific applications for customers. You never worked in applications until now.

DS: The challenge I've got today is not so much the applications themselves, although that certainly is a business challenge, it's the provision of software as a business for the company. When I came over I had three basic things I had to do. Thing one was to turn the software group into a business entity, one which made its own profits for the company, stood on its own feet and helped the company succeed as opposed to leveraging hardware.

BL: In the past, existent or nonexistent?DS: In the past, by and large, software was there to help sell hardware. So when you had a new hardware platform, you automatically put software on it and you tried to enhance the sales of that hardware and you didn't think about software as a business so much as you thought about the hardware business with software thrown in. So you didn't have to justify the software by itself, you justified it in context to the hardware.

BL: Why the certain software products that Digital has made, why they were made. How were decisions made as to what software products to create, what not to create.

DS: Well, basically the decision was, you thought about who would buy the hardware and what they wanted to do. And then you said, - well I'll have to supply them the things necessary for them to do that because either - in those days there was no third party software business to speak of - today there is a very vibrant, - you have Microsoft and you have, so.

BL: I said most of our sales were OEMs but did the software.

DS: They do software but they were not building software for multiple platforms. So when I say no software business, there were no third parties who only built software that ran on a bunch of platforms. That's a relatively recent phenomenon because that started off really with the PCs explosion. So in the early days you did indeed, you had a big OEM component. And you went and built for them and you built for specific customers. But the OEMs by and large didn't want your software at all except the basic operating systems. And the customers did and that's where you decided. But you never went along and said, - where could I make a software product that would make a lot of money on multiple platforms? So the difference between a software business and the traditional business is, - in the software business you'd say, - if I had this intellectual property I generate, so-called software or the designs of the software, how can I make the most money? And the answer is, - you put it on the most platforms. Whereas in those days it was all around, - how can I maximize the combined sales of hardware and software. And it doesn't matter whether they, quote - buy the software or not, as long as they buy the hardware. And that's a lot of bungling of the decision. Now if you look at what we're doing, we're making a business, that means each piece of software needs to look at its own business plan and say, - which platform do I run it on? How much money will it make? How soon will it break even? And, that's different than maximizing the hardware units, as for example, Alpha(?) would like very much to have all of our software run on Alpha as quickly as possible and that's a reasonable goal. However, that's in conflict with making those profits today because you don't sell any Alpha today. So you have to balance. And that balance is different than what [UNINTELLIGIBLE] .... there never was a need to balance before. The _____ group gave you a budget for your software and you did as much as you could. It's a tricky change. That was one big thing to get that business planning process in place, get people to recognize it. And that's characterized by the so-called, - Dollar A Share - campaign. I'll show you an example of one. All the employees in _____ have gotten one of these, certificates showing the management team holding a dollar. You can see they gave me this one and so that's really just an example of helping people to remember that we're in the business to make a profit. The other two things I came to do really were to set up the software architecture, the plan for technology which has been done with the so-called, NAS(?) architectures, ... document describes that people are beginning to apply that in their work. Finally to set up the engineering process so we can deliver faster.

BL: Let me go back to that architecture. In a way it seems curious that such a thing never existed. For example, there was never a common user interface for Digital systems throughout the seventies and eighties although defaulted to VMS but that's not a single user interface.

DS: What happens is that every group tended to want to be self sufficient and every group was associated with some kind of hardware and those two as appeared, decide what they were going to do but they didn't require the software except for VMS itself which was very centralized, didn't require consistency. So for example the database software we started off with ___ people, the CP software we started off with nine thousand people. And those didn't have anything much to say with the VMS people who were over here. So now we've got all of the stuff about the operating system basically connected. Now we need to have a structure that says here's how they're going to work together. And more importantly they have to build on each other's products so you don't have to write everything again every time you make a new product. So the NAS(?) structure says, what things should you count on having available? And the NAS packages(?) were about to announce, give a certain level of capability on top of either VMS or Ultrix and make it the same so the _____ parts can count on that capability being there. That's what we're trying to do with the DEC _____. We have a lot of people involved inside and outside my group, get the best methods we can. In fact you talk a lot about the ___ ____ the box that shows how you can get all the different nests(?) the basic attributes and what it says is - if you set the dials right and you turn the dials and the dials will tell you how much it's going to cost. The cost just keeps going up until the customer can't stand it.

BL: I've seen the graphic version of that but never in action. That's great.

DS: The marketing group put that together for me and it really does have just the right characteristics I think. In its full blown, you've got the nest(?) software coming out of the side, then you have a couple of other little pieces, ... put those on the top and out it comes. It's been very clear to me over time that you can't get very many people to follow what you're doing if it's all abstract. You have to provide them with something physical and visual to fasten on to in order for them to appreciate. This picture has now been translated into different languages, both National de Paris has recently used this picture, translated into French, so, for example, visibility(?) comes out _________ and they use that for that architecture. When they present it to the board, that's there. And I figure that's the ultimate in flattery for having gotten the idea right.

BL: What do you think Digital's contribution to instrument(?) technology has been?

DS: Taken in the large, I would say that we, of course, invented the concept of personal computing. The mini computer was invented to allow scientific researchers and engineers to have personal access to computing capability. Up until that time, everything was in the glass house, surrounded the priests of computing in their white lab coats. So I think that was the most fundamental first change. We took computers and sold them. The firs mini computers were sold were a hundred thousand dollars but those were very cheap by comparison at the time. Since then, we were the first to really make time sharing work. The DEC-10 for example was the first really, really successful time-sharing machine. From a technology perspective, the PDP-11, we were the first people to really start open systems. The openness of the hardware system in any case with the UNIBUS and everybody being able to make their products work on it. We had a tremendously successful OEM business because of it and there were PDP-11s working happily in the midst of all kinds of environments, telephone switches and whatever. So that open systems orientation had a significant impact. With the VAX we created the most significant scaleable operating system and computing architecture in the world, scaleable simply means it goes from very small to very large. And we designed that to last for at least a decade or more and it has lasted since (how long has it been? '80 was the first one out?}

BL Yep, architecture is '76.

DS: So it's pretty well structured, ten years and we've run out of ____ bits. We had set it up to go to as far as four billion bites of memory and now that turns out to be just, (four billion, four trillion) four trillion bites of memory, 32 bits. That turns out not to be quite enough. So we have to go to 64 bits eventually and Alpha will do that because I think it was clear that we had this family. For a decade we were successful because we had a family of computers that could provide everything that anybody could want in the way of today's open systems. As long as you stop(?) with VAX ____, it was portable between them, it interoperated due ______. The networking was fantastic. The scaleability was amazing. So we really had a tremendously powerful concept there, which is now being duplicated in part by the open systems movement. They're trying to make that work across multiple kinds of architecture. The NAS work we're doing now, the software work, the network applications support work is, in one way, just trying to duplicate exactly what the VAX VMS system did in its own proprietary environment on top of multiple platforms, called Operating System II(?). NAS(?) we thought of as providing VMS class support to all the popular platforms. If we pull that one off that will be seen as yet another major technical contribution in the world. If you talk to idea technologists, they say, - that's a fantastic concept but they'll never pull it off. And I think we have a really good chance to do it. So I think that will be the next major step. The Alpha program hopefully will be significant. My suspicion is that the commodity pressure on Alpha will be so high that it will never be a high profit margin activity although it may well generate a lot of profit for the company. There's too much commodity pressure there. The next blaze of business glory, if we get it right, will be around this middle ware software and it in turn will be supplanted eventually by the systems integration business and then the software utility.

BL: Okay. Good. Words of inspiration.

[MISCELLANEOUS CONVERSATION].

BL: Can you think of additions.
DS: Well, let's see, there's Bill Heffner, guy who ran all of Delagi, I think he's out on the West Coast at a university there in California. He ran all the software, ____11 engineering hardware while I was running all the software. He's in one of my packets(?). Bob Ray(?)

BL: In the hardware ____ team a lot of rivalry between the 10 and 11 groups.

DS: Tremendous, classic example, the 10 people were looking for a console for their PDP-10. [UNINTELLIGIBLE] ... you need a computer to do it and that was setting up the lights and they needed the left class(?) computer to do it. They decided the PDP-11 was simply not good enough for their needs so the designed another PDP-11 class machine to be the console for their ____ instead of just taking the 11. Classic nonsense.

DS: That was fun.

BL: Thanks a lot.

[END OF INTERVIEW WITH DAVID STONE].