Oral History of Hasso Plattner

Interviewed by:
John Hollar

Recorded: April 12, 2012
SAP Offices
Palo Alto, California

CHM Reference number: X6480.2012

Editor's Note:
The original transcript of this oral history was edited by the SAP Global Communications department prior to its final publication.

Hollar: Researchers often come to the museum to find out more about how someone gets started. Someone who's been transformational in your career has to have a beginning. And so, let's start with that beginning part. I know you said you didn't want to talk too much about the history, but I would like to touch a little bit on your career at IBM, your decision to leave IBM, to set up SAP, and being an almost Silicon Valley-style entrepreneur in Germany in 1972-73, I think for researchers in the future that's going to be a fascinating thing to read about and isn't covered much in the other oral histories.

Plattner: Well, it's not such a mystery because things happen in life. So, yes, I joined IBM as a systems engineer. I had been there for two years when my colleague Dietmar Hopp called me one evening and asked me to stop by his office because he wanted to talk to me. And I said okay, fine. It was past eight o'clock at night. He said there was an opportunity for us to develop a system for ICI [ed. note: Imperial Chemical Industries]. At that time, IBM offered free consultancy services and he was the lead consultant to ICI. So we had the opportunity to build a large system, and he asked me to be part of that project. I said I would be interested... I didn't realize at the time that I was signing up for something akin to the Navy SEALs. We worked very hard. We built what was in those days the first real time online sales system. That was the beginning.

ICI was a fiber manufacturer. And they were selling all kinds of variations of the same polyester fiber. And they had 50 different types of orders. We built a system in nine months. This integrated online real-time, auto-processing system enabled ICI to change orders on the fly. Reaction time was much faster for more complex queries, such as the need to address changes in the quality among different batches of spun nylon. And we solved a few other problems. We were very proud of the system. It got a lot of publicity in Germany within IBM because it was a real-time system. So we knew that we could build systems. Then ICI asked us to extend the system beyond the functionality of a sales system. So we asked ourselves the important question of whether we would do this as IBM. Or do we do this outside IBM? At first we didn't want to do it outside IBM, so we asked IBM whether we could develop it as an IBM product. And then, we were told, "No, no, no. Product development is done somewhere else. We are a field organization here." There was some product development in Germany, which was a strong region for IBM in those days, like the U.S. and Japan. So we then met the people who were responsible for standard software application development in Stuttgart. We were in Mannheim. After the meeting, Dietmar Hopp and I looked at each other and said, "No way. We can't work with these people."

Hollar: What was it about them that made that impossible?

Plattner: That is a good question. They showed us a sales chart that probably measured ten to sixteen square feet. In the bottom right corner, there was a square called sales statistics. We talked about the

architecture of the system and the blueprint of the system, since we had some knowledge of sales. When Dietmar asked the group specific questions about the sales statistics system, it became clear that they had no clue what they were up to or what they were building. So, we asked our boss whether we could found a new group in Mannheim to develop the system for ICI. We wanted to be part of this group. Actually, we wanted to run this group. IBM said no. Twenty-four hours later we said, “Okay, then we leave.”

Hollar: So, I think the modern term for what you wanted to do-- a word that’s been applied to that-- would be skunk works.

Plattner: We knew we could build large systems. We wanted to build the real thing. We wanted to build an application that could handle more than one customer. This was 1971. In the early 70’s standard software started to pop-up. And this was not only the systems software, operating system, databases, and things like this. It was now also applications. And since we were between standard software-- systems software and applications, we realized that we could do both. And we had proof since we had done one in nine months. And then, IBM said no. And the potential customer for the prototype was still waiting and asking, “What’s up now? Who’s doing it?” So, we went back to the customer and said, “We are doing it.”

Hollar: You were very close to a customer who wanted that at that point?

Plattner: Yeah. It was ICI, Imperial Chemical Industries. They were, basically nurturing SAP. We were living there. We got our desks there. We ate in their cafeterias. So, that’s the whole thing. We became part of this company. And this was a prototype for standard software that we would build in the future. We developed a system for a multi-national, multi-site company requiring multi-lingual, multi-currency processes. This gave us a huge head start in dealing with other international companies. So, from Day One, we did not work for a solely German company. We worked for an international company, which was in those days the largest chemical company in the world.

Hollar: As you look back on that, does it seem unusual to you that you had the courage or the will to leave IBM and to start out on your own with this small team?

Plattner: It would have been difficult without the success of the real-time sales system we had built previously. That was basically the backbone of our decision-making. We knew we could do this. But then, we didn’t think much about the consequences. There was a very interesting project. We never talked about potential failure. We knew we were in a very new discipline of the computer industry. We were doing applications with terminals. We were doing remote processing. We actually did real-time processing. So, we were absolute leading-edge in technology. And we were not afraid. But we never talked about this. We all knew: failure was a not an option.

Hollar: The relationship you had with ICI also allowed you to self-fund, I guess, at that point, right?

Plattner: We were profitable in the first year.

Hollar: That's amazing.

Plattner: That was actually probably a tiny little story. I think it's in this one.

Hollar: It is.

Plattner: It is. After reviewing our contract proposal, the local board of ICI and a representative of Manchester, who was overseeing European subsidiaries, called me and Dietmar in to meet. They told us that they would not sign the agreement because the price was too high. We looked at each other, left the room and said "Mm-mm, the price stays." We went back in and said, "No, that's it."

Hollar: And did they accept it?

Plattner: And he took the pen and signed. "We were just kidding," they said. So, that's how it started. So, I never understood, for many years, that you have to give discounts and you have to negotiate. You have a price and -- so that was for many years, the way we operated. This is the price and -- sometimes our price was wrong, and then we had to adjust it carefully. And always thinking about whether we were making a mistake, considering that this put previous buyers at a disadvantage, and things like that. But we never gave a discount. That changed later when we made much bigger contracts, and different types of deals.

Hollar: And with respect to the software, in particular, at that moment, what was it that you saw in the immediate future of software development that made SAP's approach unique?

Plattner: We believed that most business processes we knew about, especially the whole world of transactional processing, could be done in real time. We would not artificially introduce a batch process because it takes too much time. We chopped the batch processes up and let them run in real time-- in quasi-real time, so that we reduced the amount of batch processing to a large extent. Later, we created new batch applications. So, we were not yet 100% successful. But all data entry was 100% real time, which led to a system in which once data has been captured, you can rely on the data. A multi-step process was no longer required to validate data and accept data. And then days later, or probably weeks later, you can really rely on it. When data was in, one can actually work on it. This would make it possible for business processes to follow each other instantly. So from order intake one could instantly start to

produce the order and ship the order. And this could be done as quickly as manufacturing, transportation, and all the other steps in this process allowed. Software would no longer hinder the business processes. That would come many years later when we came to the U.S. In the '90s, if I may jump forward, auto-processing time was in the neighborhood of 60 days between order-taking and delivery of a digital product. So that was our claim to fame for the first year. Second was integration, that we integrated multiple disciplines. First of all, everything within an accounting system, then sales and accounting, then purchasing and accounting. So, we built systems which know each other, and that was also relatively new to the world. Systems were isolated partial solutions. And they were stitched together, typically, in a company-specific fashion, per company. So that was the pitch for the first decade or more. It is real time. It is integrated. Whatever you put in the system, you can instantly look at it. You can control it. You can proceed in the business process.

Hollar: Some start-ups like yours master the technology or the software, and don't master the business of being a business. Some have a good business model, but a product that isn't any good. How did you manage to squeeze both ends of the balloon at the same time?

Plattner: Definitely the IBM education helped here. So, I was an electrical engineer. Dietmar Hopp was an electrical engineer. Klaus Tschira was a physicist. Hans-Werner Hector was a mathematician. Claus Wellenreuther was an MBA. But we all worked for IBM. I worked only for three and half years, the other ones a few years longer. So, we got an excellent education from IBM. And IBM is a thoroughbred business-oriented enterprise. And we understood how to make money, how to negotiate, how to keep track of costs. So that was, from the beginning, part of the genome of SAP.

Hollar: And was IBM's attitude, by the way, as you all were leaving to do this, was it "good luck"? Was it "you'll be back"? Do you remember?

Plattner: My boss's boss was in shock because he had to answer questions such as, "How can this happen that five guys of your local operation are leaving? This is not good." I had just been promoted and was slated to become the sales executive for BBC, one of the largest companies in Mannheim in those days. And suddenly, before I even took the new assignment, I was gone. So, they were in shock. There were some expressions of hostility, but no one acted on it. In the end our boss said, "Guys I'm always with you, whatever business you're in." And they still viewed me as a member of the family. But officially IBM saw us as a coming competitor. And it took some time before we worked cooperatively again.

Hollar: Why did you grow so quickly?

Plattner: We didn't grow that quickly. Actually, that is-- we grew from the Internet company perspective, very slowly. I still remember that on my first trip to the United States in '84, we had $33.4 million in

revenue. The dollar conversion rate was very high, but it was under 100 million Deutsche Marks, I think. So we were a relatively small company back then. We had probably 200 customers. With R2 we then had two and half thousand customers. So it was not that big. I think when we started R3, we were somewhere between 200 and 500 employees. So, we did not grow that fast. First of all, the hardware had to grow. Our system worked for somebody who spent a little bit more than for a normal enterprise computer system in hardware. The dumb terminals were pretty expensive in those days. We needed a larger computer. So, the large companies could not jump on SAP. And the very small companies could not jump on SAP. So, we had a small segment of aggressive companies who wanted to have a modern system and could afford it. And that changed when IBM came out with new computers. NEW mainframe computers shrunk, became much smaller, and smaller companies could run a mainframe, as well. That brought about the success of R2 --tenfold the success of the first software suite.

Hollar: And were you anticipating that was going to happen? Were you looking at what was happening in the industry and saying there's better hardware coming, there's going to be big improvement here, we need to be ready for that?

Plattner: I would say until the late '80s, probably '88, we were completely living in the IBM and IBM-compatible world. We did not see the world outside IBM. So, we saw IBM and we saw Siemens as an IBM-compatible German manufacturer. We did not see any of the other, I think there were 12 other hardware companies, computer companies in Germany, [like Nixdorf Computer, Kienzle Computer], you name them. We didn't see any of those. We didn't work for those. We purely worked for the IBM world. IBM announced the SAA system application architecture in late '87 or '88, I think. Klaus Tschira and I went to the first presentation of the ideas. On the drive home, we said, "That's it. Now, we build-- we have to build the next version of our system." We had survived the first version. For a software company, the transition from the first version batch of a product to a second is difficult. And it was for SAP. The next day we started to gather what we needed to build the new system. We were programming in C. We discovered that IBM had a great idea, but did not have the computers for it. So we began to looking around. At the time the microprocessor was really catching on. One stream was going into PCs and the other ones were going into workstations. There were different chip sets -- one was the Intel world and the other was RISC-based instruction. With reduced instruction-set CPUs, there was suddenly an unbelievable level of computing power that we mainframers could look at. And so, in this process of building something for IBM SAA, we switched from IBM to client-server non-IBM because we saw more of this architecture outside IBM than inside IBM.

Hollar: As you look back on this now, and you're still experiencing it today, too, what's the key to managing an inflection point like this where you're going through these major changes, and you have to be so nimble?

Plattner: Sounds crazy, but you need hands-on experience. Siemens also struggled with that. “Where do we go in the future? Do we go with Intel? Do we team up with HP? Do we do Silicon Graphics International—(SGI)?”

Hollar: We’re in their old building.

Plattner: Yeah, I know. I know, yeah. “Which chip technology do we go with,” they asked me, and I said I didn’t know. But I can tell you-- we experimented. So, we tried IBM RISC computer. It was so slow that we couldn’t even get the hello, good morning message up. So this was not working. And it was expensive. Then there was the IBM UNIX computer. But they had C-Programming. Then HP asked us, “Can’t you try us?” And they were not so exciting, but we got a few HP machines and played with them. It was a very critical moment in the project. In the R3 project we were developing, we still wanted to develop for the mainframe, the next version of the mainframe computer. But we didn’t get the stuff on the mainframe.

So, we ventured out in UNIX for development. This was not very promising. And I was pretty depressed. Dietmar came in and asked, “What do you think about UNIX?” And I said, “UNIX is nix.” And he said, “Think about it twice,” and left the office. So, I’m sitting there asking myself, “Why did he say that? Did he mean that we probably should not build for the mainframe? We should build for the UNIX system? Is this an alternative?” And I said, “No, no, no, no, no. Get this thought out of your brain.” Then we got a DEC machine, 25 million operations per second. It was a little pizza box with a screen on top. And we tried to get this prototype up and running the new SAP system written in C, with an SQL database —and a graphical user front end. When I left work at probably nine o’clock, there were 18 people standing around this little DEC machine. And it wasn’t working.

Hollar: And you declared at that point--

Plattner: I declared it’s over. This is not working. The Hanover IT fair is coming. And I didn’t know what to do. So I went home and came back the next morning. And there’s a note on my desk. “We got it working.” I was not a technology guru, but all of a sudden we thought UNIX was probably the way out -- out of the problem. And HP was saying they now had not only a 25 mips machine, but a 50 mips machine. And I said I need 200. And they said what? “We can’t give you 200.” And I said, “Please give me every single machine you have. I need 200 in six weeks.” And we shook hands and we got the 200 machines. And this was the break through. We were over 1000 people at SAP with over 500 developers and 200 worked in the R3 project. And everyone now had a 50 mips machine. This was the supercomputer age. Within six weeks we wrote the software to synchronize the 200 machines so that we had a shared development system. And that was the real beginning of R3.

Hollar: So you started that answer by saying you have to be hands on. That’s how you get through these inflection points.

Plattner: The hands-on experience of the DEC machine was “wow”-- this is the future. A compilation of our basic system takes only an hour, and then we can continue and not wait several days. This was very important for our developers. We got a C-code debugger. So while we were testing we could debug what was happening. So we were much faster all of a sudden in our development. When we started the debugger on the mainframe, the administration complained. Sales said they could not work anymore because whenever the debugger started one of the big IBM mainframe processors stopped working. All of this was new for all of us. IBM looked at us and asked us what century we were living in? And so, we jumped one generation forward. IBM had difficulties making that jump just a year later. And they needed Lou Gerstner and the big, big, big changes in IBM to cope with the changing world from mainframe to desktop computers. But we jumped, basically, overnight.

So I said to my colleagues. “Forget about what I said yesterday – UNIX is nix. It is only UNIX. And we will not go back to the mainframe.” We started developing now on a huge distributed development system. When I met Scott McNealy at the Hanover fair that year, he told me that Sun - the Stanford University Network - had a client-server system. And I said, “Sorry it’s too late. We have just done HP. We cannot do Sun as well.” He has never forgiven me for that decision and we in fact did develop on Sun much, much later.

Hollar: If you’d met him earlier, would that have been different?

Plattner: There were people already lobbying for Sun. Probably for me from IBM via DEC to Sun was too much of a radical step. So, HP was a company I knew as an electrical engineer from the beginning. And the machines looked good. And they had good specs. So, probably it was just a matter of history I met HP first. It was this famous Hanover IT fair where the R3 story started.

Hollar: It seems to me in the story you’re telling, I’m also hearing you have to have the ability to let go of something you’re committed to in order to pivot and change, and move in a new direction.

Plattner: That’s probably the other part. This was hard. But you have to be true to the facts. We could prove that the prototype of our new system was effectively too slow for an existing mainframe. It doesn’t help to bang your head against the wall and say we tried to make it work and it didn’t. . And it is not easy to admit you failed to develop software for the AS400. And you did not develop under SAA software for the mainframe, at least for the first five years of the ‘90s. We did not succeed. But we had a tremendous success on the UNIX networks.

Hollar: And how do you bring a whole organization with you when you are making that kind of a strategic decision?

Plattner: We didn't. We didn't. We split the organization. And this is a hard part. It was not only the split of the organization. It was a split of the market. The official SAP believed in '92, late '92 that the peak of the mainframe sales will come in '94, '95. And I said, because I made this transition to UNIX, I say you won't sell anything anymore. So we did something which is hard to do. I went to America to sell R3 because America was much better prepared. If there is a new wave, Americans -- or a large portion of this huge country - is interested in the new wave. Germans are a little bit more conservative. They come a year or two later. So, we got customers in Germany, but not enough to make enough money. So, I could only make the money in America.

And then, Dietmar and I made a handshake agreement. Dietmar would continue to run R2 and get as much out of R2 as possible in order to finance my R3 adventure. It was clear that if R3 failed, we were done. But even if it didn't fail, we would have a transitional period. To mitigate the transitional period, the pangs of that, we sold another share of our company, which was in vain. We didn't need the money. That was bitter. But it was an insurance policy. So, we sold, through Goldman Sachs, a significant stake. So we had the cash and the cash was basically for one year. We can run the company for one year. We didn't really think it through. You can't run a public company with zero revenue, even if you have the cash. But we felt better, and I did not think about financial crisis for the two ramp-up years of R3. I knew we had to hurry. But soon you could see the potential. And then all hell broke loose and SAP grew 60% per year. And the rest is history.

Hollar: When you look back on that decision, does it seem as risky as it sounds now when you're talking about it?

Plattner: Let's say I'm now chairman of a 55,000-person company. And they come to me, and tell me that they are taking this risk. I would take a deep breath and ask, "Do you really know what you're doing?" They tell me, "No, but we have to do it." We might come to this and actually SAP is at a similar crossroads when you follow what we are doing now with on-premise and cloud - a very similar situation, very similar situation. Very risky if you jump too early, but it's deadly when you jump too late. When you're a little bit too early you run out of steam because the technology you need is not coming with you fast enough. If you jump too late, history is brutal. Jump from one system to another, from one world to another world. So it was risky, but I think it was the absolute right decision for the company and for the product.

Hollar: In the middle of all of this, the Internet is also happening to some extent in this period that you're talking about.

Plattner: A little bit later. My first exposure to the Internet was when we had an American long- haired California dude working on the R3 project. So, the first thing he tells me is he doesn't know how long he will stay with us. I said, "Oh that is interesting. We [are a] German company more in the Japanese tradition. We are lifelong employees." The guy says he is with us for now since it is fun. Then he asks whether he can bring his bicycle to the office. We were already crammed in one large office because we

were all sitting together. We took large offices so that we could talk with one another. And he wanted to bring his bicycle in. He said everybody in California brings their bicycle to the office. I said, “Oh God, there are more things to learn here.” And then later that evening as I was leaving, he was sitting there hammering into a terminal. When I asked what he was doing, he said, “I’m on the Internet.” “What are you doing on the Internet,” I asked. “I’m asking my guys in California what we do with, for example, a fork process or do they have any idea how we can improve the make time,” he replied. He said he was talking to people he knew from university or in other companies. “We share our problems and thoughts on the Internet,” he explained. “I give them a question, and when I come back to the office tomorrow, I have an answer.” This was the beginning of social networks. He said this system was originally developed by the U.S. military for research and was not available to use. “It’s the future,” he said. “Okay, but its characters and character input,” I said. “And so it is basically a typewriter protocol.” “Yeah, but it’s worldwide,” he said. This was in ’91 or early ’92. And only a few years later it becomes the network-- how we connect the front ends to the application servers. And then, 20 years later, it’s the backbone of basically everything that we do.

Hollar: Was he the one, who gave you the advice, don’t forget about the Internet?

Plattner: He always said, “If I leave, don’t forget the Internet.” So that’s the point. He said this. And then he left. He left. So, we had to find somebody else who maintains our IP address and everything, and so on. So, that was the first exposure to the Internet. And then, not much later, so it’s in ’93, I started development in the U.S. because you saw instant interest in the U.S. for our R3 system. Companies were instantly interested and enthusiastic about it. And then I remembered something I had learned from IBM -- If you want to be successful in a country, in a large country like the U.S., you have to have development there. You cannot just export from Germany into the U.S. You have to come to the U.S. and do something there to show commitment, to show customers this is not just an export item here. So, we manufacture locally, as well, even if we start small. So that worked. We have done the same thing now in China, and it works. We have done the same in India. So the Internet was there. We had a worldwide communication system. And I wanted to enable developing 24 hours around the clock seven days a week. And I tried to start this, and that failed. Later we would come to HANA – and then it would work.

Hollar: Do you recall the moment, from a business perspective, where the Internet became something that was just out there to something that you said, “We really have to know more about this”?

Plattner: Yeah, yeah, yeah. So it started in ’96, ’97. The journalist, “What do you do on the Internet?” And we said that we connected our front ends over the Internet. We were still so busy developing missing business functionality -- ERP functionality -- in the system. Despite our financial success and growth, we didn’t have enough time and energy to think about real alternative applications. We did this only in I think in ’99 -- first in American markets and then later with portals --separate entities doing completely different types of applications. We enjoyed relative success and, after a short period of time, we were market leader in marketplaces. Unfortunately, marketplaces didn’t turn out to become a large application area.

They're still there, marketplaces, but they either became gigantic marketplaces or specific ones like eBay for trading goods between people-- and companies. But our idea to have professional marketplaces for companies did not work. We made a logical mistake. All the participants have to agree to share this marketplace.

And I went to a meeting in Tampa Bay, Florida, and most of the companies represented were SAP customers. Only one was not. The other SAP person at the meeting told the group, "Since you're all SAP customers, it's the natural thing now for you all to take the SAP marketplace offering." The group discussed this when we left the room. The one non-SAP company – a well-known U.S. consumer products goods company -- turned us down. The company was not interested in this "all SAP or nothing" approach and selected the competition instead. This company was right in deciding that: "If SAP treats you guys better than us only because we run an SAP competitor's product, this will not work." It took us a while to understand this position. You have to understand this-- if you are the dominant company, why should you not dominate. So marketplaces were the first area where we saw the need to define a policy of openness and choice. Open for everybody, even open for other competition.

Hollar: And that was the Internet effect, this openness?

Plattner: That is the start of the Internet. The Internet is the technology enabler, but it's also now the mindset. Now, we do something together, but if we do something together, we all have to be treated equally. It's actually logically, instantly understandable. I wasn't smart enough. Others were not smart enough. So, it was not number one on our bullet list. Our marketplace is open to anybody. We thought if we have half of an industry, in this case it was the consumer goods industry, and then we should be the natural selection. Why are they not doing this? And we didn't understand this instantly. Then I realized that we had missed something here.

The second obstacle to offering alternative solutions was that the sales force could not cope with selling this way. It's very hard if you have products that overlap in some areas and are open to others. We are learning this now with cloud and on premise. But you have to be smarter. We saw this with the portal. The owner of a smaller company in the Bay area wanted to run a portal on SAP. One SAP salesperson said that the portal and the ERP system should be one system. But the portal salesperson told the customer that the portal would also run on an Oracle system and that it would be fine to buy Oracle. That killed us. You cannot have infighting in the sales organization. And we failed. We didn't get this customer. And so, I gave up. That Easter I folded markets and portals back into the main SAP business. And they became, then, SAP Technology and part of-- that's the starting point of the fourth version of SAP ERP systems, the MySAP and later SAP Business Suite.

Hollar: What year was that when you made that decision?

Plattner: I think 2001 or 2002.

Hollar: Yes, that's what I was aimed at-- 2002.

Plattner: 2002. Another inflection point. Something is different now. We cannot become an Amazon. How can you come from an ERP company to be a bookshop? Or the closest was probably eBay. But we didn't become an eBay. We were not consumer-oriented. But now you can already see the big swing toward consumers. The Internet is the connection of consumers to systems. Prior to this, we lived completely in this corporate world. Now, the corporate world is totally linked to the consumer world, these other portals. This is sales on the Internet, Internet presentation of products and interaction from consumers to business. So, the B-to-C process now starts proliferating and exploding.

Hollar: And this was another-- it seems-- well, when you merged portals and marketplaces back into the main company that was an embrace of that, wasn't it? It was this whole shift from B-to-B to B-to-CC.

Plattner: Yeah, but now we fall back. Now we have them back. And we instantly then use the portal for corporate purposes and not as a sales front. So, we missed-- we didn't build real good software for Internet sales, while there were probably hundreds of Internet sales companies and products for different industries, for different purposes popping up. So we did not really embrace the C. In business-to-business we tried to hang in and do as much as possible, but we let the end consumer go. And that had an effect in the following years because once the end consumer was able to work with companies over the Internet in a different way, employees started to think, "Why can't we do this too?" So this is the beginning of employee self-service and working differently with systems. And now the struggle begins over how much rigidity, control and governance you need. Or it is just an easy system that you can learn to handle in five minutes.

And now everybody is doing it. You can book tickets over the Internet. And you can buy and sell over the Internet. You can bank over the Internet. There are warnings that this is dangerous and something could happen. But it happened before. People lost checks. Or people forged checks. So, it happened before. So now fraud just happens in a different fashion.

So we had a setback there with markets and portals. Dietmar, who was already on the way to leaving the company, told me, "If you do this, if you fold them back in, you can never get them out again. You'll miss that chance forever." I said, "I know this, but I can't continue with this. This is not working. They're killing each other, the salesforces and overlap in functionality, etc. etc." A little bit more design thinking process here would have helped to come up with a better solution. So we lost some time and have to pay for it now because now we have the next wave already. And that is the on demand, the cloud solutions, which is only a continuation of this process. So, we now become directly connected. The C people, the consumers, get directly connected. The consumers or people in general are building networks. They're

communicating over networks. They become independent. For the last three years, I have bought everything I buy over the Internet. I have been at the car dealer only after I bought the car already, only to see it - a courtesy. No reason to go there. Courtesy. And for years now, we are on the way to build applications for the cloud. And here, had we handled this transition better we could have understood earlier what's so attractive in the cloud.

People say it's cheaper in the cloud, and we can use hardware differently, and so on. The classic IT cost of ownership, total cost of ownership argument. It's not-- this is not the important thing. The important thing is that you can build applications specifically for one process, for one-- some steps of the large business processes, where you can really do a specific service. And the integration of these services has to take place. But you do it only once. And you do it once, and everybody is using those. And if these processes are not working this way -- if they have to be customized too much, then it's probably not working. But for many processes, it is working. And typically they are less complicated because they are spot solutions. They are not generic solutions for everything.

SAP is not the only company which has the technology for integration anymore. Yes, we did the largest integration twenty years ago when R3 came out. But now everybody has learned how to integrate. The whole Internet is permanent integration between different applications, different systems. The world has moved on tremendously. So integration is not a criterion. This alone does not work. The attraction is that it works with just a click - nearly like an iPhone app. I click, and I have it. And I click and some data has to come into the cloud, and then they tell me I have to ship this data up. And then they can start working. And I have talent management.

I take one of the products we have now acquired with SuccessFactors. So comfortable, they're so easy. And this will be the driving course. The cost, yes, the provider of those services has to figure out how we make money with this. But like in many of the other Internet-based enterprises, the user numbers are skyrocketing-- SuccessFactors is approaching 20 million users. You do not have to be at 30% profitability from the start. So, go, go, go, go up and expand your user community - many technology problems you can solve later because if you have such a large community, this is the gravity. The community is the gravity. The people who like to have these simple apps. Nobody will give up the enterprise backbone. Ten years ago my friend Scott McNealy said everything will be in the Internet, and we have the Internet applets. And they will be all interconnected. And so, there's some truth. But we will not give up our enterprise systems. Partially, yeah, but probably not in construction. Probably not in manufacturing. Probably not even in financials. And if you give them up in financials, then somebody makes a concise decision. It's equivalent to my Sharks here. I'm a co-owner of the San Jose Sharks. They run on our on-demand product. And they actually like it better because they have less hassle. There are a hockey and entertainment company-- hockey club and entertainment. They don't need to deal with the complexity of an ERP system. So, it is the ease of use, the user experience, and the unbelievable potential of the growth rate. If you do something right, boom, the numbers come.

Hollar: You and I are actually doing something right now that the museum has long dreamed of, which is to take someone's oral history, and then revisit it 15 or 17 years later to talk about lessons learned.

Plattner: This is dangerous. This is dangerous.

Hollar: And I'm loving this.

Plattner: This is dangerous. Yeah, but you see I have to correct myself. And it's not always that you see this. It's easy to write an essay, I see all this and then you can even write multiple ones to cover the variations. You make mistakes. And you said it before, the hard part is to realize it's not good enough, or it's even a mistake, or we have to diverge. It is only after acquiring SuccessFactors that we decided to write a second layer of enterprise applications on top of the existing enterprise applications for the people who do the normal stuff. Not every purchase is a turbine for a nuclear power station which has to be super designed and checked and meet the highest quality and safety standards. There's a huge process involved in this. There are things like I'm just looking at a telephone system. Everybody should be able to decide what telephone system is in this building here, as long as it can connect with the other telephone systems in the world. That's the only thing. It has to be a telephone. If it's a telephone, it's good enough. Nobody has to do this in a super central way anymore. Much to the chagrin of the old telephone companies, but this is how it is. It has changed. So we think about and started to build now applications of a completely different type. They can define-- like start-up companies.

Hollar: I was about to ask you about that.

Plattner: They're small in scope, and there's only one unit of measurement, users. If you get users, and if you get thousands of users quickly, then you can create a new business opportunity because you create gravity and momentum. SAP did an application here Recall Plus. It's pretty new. It's just breaking through 20,000 users now. You can go on your iPhone, it's an iPhone app, and check in various categories, there's a central register in the U.S. of which products have been recalled. And with click, click, click you are at [baby]buggies, and which ones -- and there are the photos of those provided by the central register which are under recall because the wheel breaks loose or whatever, or they have a sharp corner, or recalls of pharmaceuticals, etc. Twenty thousand users, no advertisement, zero, nothing. It is just word of mouth. People are asking "Have you seen the recall?" One mother says to another "I don't have a lot of free time, but I know now what could be dangerous. And I'm telling you, you should have this app. I know because I have this app." And - boom there's the next one.

So, we want to build applications like this. And not leave the whole field to the start-up companies because we can-- our people can do this, as well. They have fun doing this. And you can learn. For example, we show the recall application to one of our twenty-year customers and he or she says, "This is exactly what I need for my product." We had not previously considered this as part of our normal product

sales service. If they had asked us to develop this, we probably would have done this, but they don't think that SAP does things like this. And you have to do it using a different technology. You have to do this in a lightweight fashion without allowing it to become the center of the service system, or the call center. It is a tiny little app, which has its own value. Otherwise there's no use. It shows there's value. And if you need more of this, it can probably grow. Other than that, it is just this added value connecting to existing systems.

Hollar: It seems like of all the pivots that you've made, this is one of the biggest because as a software company, you're talking about an entirely-- it sounds to me like you're talking about an entirely different ecosystem for your engineers and developers.

Plattner: Actually, the engineers of this type of app, they live in a different accommodation. They live in app houses. What is app houses?

Hollar: Is that by design? That's deliberate?

Plattner: It's by design, and actually you use the word by design. You know I participated in a project here at Stanford University, the School of Design Thinking. And I was completely convinced-- I still am completely convinced -- that a very interesting and necessary component for future design of products and processes is to have a multifaceted, heterogeneous group of people working on the design so that we don't get stuck in what we know, and what we can extrapolate from what we already know. So, we have to challenge ourselves like the start-up does.

I have an idea. Can we do this? And then somebody finds somebody else. And if they agree we could do this, we try it. Design thinking stimulates us to break out, think and engineer. So app house is our mini design thinking labs. We rent separate space. It's open space, no individual cubicles, no separate rooms. The Germans live in separate rooms for three people. Microsoft lives in single rooms. This is open space. There are desks, but there is no private office, no corner offices. All walls are for scribbling and putting post-it notes on, or drawing something. Unbelievable media we have. We take an iPhone, and you already have something. Put it in the computer and you can draw about. You project it, and you have something everybody can see. You print it out, and you have a huge print out, and you pin it to the wall. This is a different world. So, in this world, they live. They have red couches. This is the trademark. Stanford started having students and teachers sit on red couches when they have a pow-wow or discussions. The app house people are allowed to have crazy ideas. And if nobody listens to them, they build it. If somebody in development listens in and wants to collaborate, this is also allowed. It's not forbidden to work with mainstream development. And we build applications like this. We have now an app house here. We have one in Ireland. We have one in China, and in India, and in Germany, the Works Council. You know what a Works Council is in Germany?

Hollar: Yes.

Plattner: This is that the employees have a right to basically check everything that management is doing, hiring new people, or firing people, or changing the control of a division, or whatever. Germany is the largest organization of SAP, which doesn't have an app house. This is a German-specific thing and will come later. And in these app houses, we hire young people. You can say and do whatever you want. But let us know what you do. And share it with others. All young people are network people. And they're all chatting with each other. There is no problem anymore now that we have the Internet, the computing speed, and the cultural change that encourages people in China, India, Ireland and the U.S. to work together despite time zone differences. They stay up a little bit later, or get up earlier. This is very encouraging, very exciting, so this is where SAP breaks out.

And yes we have to develop these applications in a different environment. We have in Potsdam an innovation lab. It is a development lab. It's not a pure research lab, but they do prototype applications or start new applications. For example, we develop health science applications like the calculation of a genome from sequencing. We believe we can speed this up by a factor of thousand plus, which would be commercially a huge improvement over the current situation. More and more companies are working on genome-based research, medication, and patient management. And we do this in Berlin together with the largest German clinic, the Charité from the Humboldt University. We do it here with Stanford with Professor Bustamante. These are things SAP didn't do two or three years ago.

Hollar: Are you trying to measure in commercial terms what this means to SAP to have something like these app houses?

Plattner: We will measure by usage. For this health research, we have to have a little bit more time. For apps like Recall or bringing sports club fans together for merchandising, we give them 90 days. And after 90 days, we measure whether there is hope. Do we get another 60? Or are you ready to roll out? And people have to learn to reduce. The strength of a start-up company is that they have to, and therefore they are able, otherwise they die, to reduce the scope of what they build. They cannot build everything. They cannot do everything they dream up. I think the strength of SAP was, when we started, that we knew what we couldn't do. I simply fought accounting to the extent that I could build it. And then I defended it and this was enough. Forty years ago, I did exactly what the cloud people are doing today by not building everything which was necessary. I said, "You can do whatever you want for your month-end closing or year-end closing, that's too complicated. I can't do it. So, do this in whatever you do it, but everything else, which is easier, I can do. "

Hollar: And that was enough?

Plattner: And that was enough to build the largest accounting system. The more we build in functionality, the more difficult it becomes. This is the hard part in a car. You have to say-- my car is not a sports car and a limousine and a taxi, simultaneously. It cannot be. So you have to make a decision despite the fact that we see more hybrids now than regular cars. So, I tried to blend. But this is always a product decision you have to make. Where do I stop? How much-- how far do I go? And the good thing is, now with the Internet, you can check. Where's my market? And if the market is there, I am so intrigued.

Everybody is intrigued by the Apple success. And they deserved it. A company stumbling ten years ago building better computers and operating systems -- an order of magnitude more simplistic than the bedrock operating system they were working on in those days, goes into the intelligent phone business. And all of a sudden, they find the sweet spot where they know enough about technology, know enough about manufacturing that they can let other people manufacture it. And it becomes a success with the users. So, I think this is very important that you have this triangular-- the Stanford design guys call it - Desirability: somebody wants it. Feasibility: it's technically possible. And Viability: you make money. And you have to have the three always together. You cannot say, we said this many times, we have solved the problem. And we can sell it. The UI, we will improve it later when we know more. We call it user experience, but it's more than just the UI. It is the performance. It is the way you can acquire it that has changed. And now, we click, and it comes down from iTunes or any other online distribution mechanism.

You have to keep this in balance. And this is hard for engineers. So therefore, you need teams with different expertise. You have to figure out how to keep one group from dominating another. It doesn't help that you have somebody who doesn't understand the business, and says I need this, I need that. Or they don't develop it because it is not feasible. Or it's not feasible under these circumstances. So that was the thing Apple figured out. We can do an iPod better than the absolute king of the Walkman. Boom, wow. The tape is gone. Yeah, okay. It's more than the tape is gone. It's you can use it. It's so easy to use.

Hollar: It's a whole ecosystem.

Plattner: And all of a sudden, there is the ecosystem coming with it. I can download. Oh, this is interesting. I don't have to go to the shop and go through the CDs anymore. And yeah, we can build. It's a completely different form of integration we can build now. And I come back to the point; it is now from the manufacturer of a core component, probably with the partners, or other core components together. You can build a system which can really bring the consumer to the producer of another product or service in real time, instantly -- and probably to move to my pet project for the last five years: HANA.

Hollar: I was just going to ask you how this has influenced the development--

Plattner: If you can now build a database, or have a database, which has the attributes of HANA, it's extremely fast and so unbelievably fast that most people think this cannot be. It cannot be that much

faster than traditional databases. Yes, you can. When you do something like what the iPhone did. We can have 80 gigabytes in a phone size. We can get this memory stick. We can have a processor which is fast enough to handle even multiple parallel processes. We can have a camera, an antenna, and we can have a little speaker and a touch screen with a fantastic resolution together in a device which is of this size here. I don't know where mine is, here. Unbelievable, there it is. It can do all these applications. We can.

And what we did with HANA is the first research project at the university. You don't have to succeed. So, I said we do a research project. We build a new ERP system. We didn't move far beyond the database. We started with a database. So, let's start with something new. What will happen, I asked my students and myself five years ago. This is 2007. The multi-core processors are coming. So, first there's a dual processor, then we have a CPU with four cores. And then we have a CPU with eight cores. Memory is getting larger and larger; DRAM memory is getting larger and larger. What happens if all data is in memory was my question. What happens to our commercial systems? We can totally concentrate on the building of functionality and not worry permanently that this is too expensive, or that we won't achieve a reasonable response time. So, in the past we always had to limit ourselves. As great as the disk was, we started with a disk forty years ago. It's the grandmother of all online systems. Tape wasn't. Disk was direct access. Memory is the solution. Thousand times faster, more than thousand times faster. So, the first question is will memory be large enough that we can really run complete systems. No, not at the moment. But it doesn't matter at the university. I said, "Okay then we only build a future system for small companies. It's good enough for research." And while we researched and researched, every year the memory size doubled. We started with 32 gig, 64, 128. Oh, we get a terabyte next year. We get two terabytes if you want. We are now at 80 processors per board and two terabytes. And whether this is the best combination is a technical discussion. We actually go for little smaller boards and have more of those. So, we started and put together different database technologies. Things which had been researched in other places or even built in products, but which we put together simultaneously. We don't do disk anymore. Disk is done. We do not mathematically think about the deficiencies of a disk. There's a paper from 2006, Gray-- the-- John Gray?

Hollar: Jim Gray.

Plattner: Jim Gray.

Hollar: Microsoft.

Plattner: Yeah, from Microsoft. Jim unfortunately died. He wrote this paper "Disk, Yesterday's Tape". Yes, 100% true - every single word. He predicts what is happening and will happen. And we just work along. We could show within a year that in a prototype of such an enterprise system, we run something like an SAP dunning process, which runs today, 20 minutes before we can start printing or sending emails or do something. Twenty minutes it runs on the SAP data set of SAP customers. And we don't have that many customers but enough that it takes 20 minutes. The students were down to three seconds.

The next batch of students -- students are building the programs and the testing and doing everything - got below one second. People were asking me why do you do this. Nobody wants a batch process like dunning to run faster than let me say two minutes. And I said, no, this is not the point. The point is that we have proof now in one application that we can run large traditional batch programs in real time. If this is true for dunning this is true for trade promotion analysis. This is true for instant calculation of pricing based on the current situation in the market. Wall Street does that every single second.

We can do things now we couldn't do before. Take the cost of distribution of spring water throughout China. The calculation takes two or three days. It's now down to less than three seconds. So the company is feverishly thinking, "what can we do with this?" We could come to a dynamic pricing depending on the traffic situation, transportation situation, we can have different pricing. So we can become much more flexible. The more information you have, the more flexibility you get. And we have one after the other, great samples of this kind of thinking. Why? Chinese companies are young and they take anything from the world and try it to apply it to the humongous volume problem they have, the huge size of their country. So they have all of the problems of the United States, plus. Therefore, they are very eager to radically think new things.

Hollar: And so HANA is designed to take advantage of this dynamic.

Plattner: Now, over the past two-and-a-half years, I have proof at the university this is the next generation. We have to do this-- SAP, or somebody else has to it.

So I said to SAP, "If you want me, I can do one more project but only under one condition: I'm a dictator. I'm a benevolent dictator, but I'm a dictator". You do what I say or I walk away and you're on your own. I cannot run like in the normal way-- you fight for it, you fight for budget, you fight for people and you fight the naysayers. I did this all of my life. I don't want to do it anymore. So either you all are believers and you follow and then you have to listen to what I say. Or the good dictator listens. If you can really convince me in an argument, then fine, I might change my mind.

And the next day I had the first issue. The database we built has only one interface and that is SQL. They had no SQL, only other stuff. I said, "No, it is SQL because that is viability." That's the only way we can make money. If SAP builds a database with a new user interface nobody knows, who cares. Who knows about file search index interfere, FSI? That was the interface of the prototype. Nobody speaks it. It is close to SQL, but it's not SQL. What is the advantage now? We take any Oracle or other application written on an Oracle database, replace the database with HANA and it works, plus/minus instantly. We replace a Teradata system with HANA and it works. Why? Because we use the same interface. I still believe that SQL is one of the best things the software industry has ever designed, probably XML and the other ones are of similar quality - and the Internet. Obviously, Internet protocol is of probably even greater importance.

So we did this. And we started working. And from a development project perspective the process was quite radical. The development groups start in the east, in the morning where the sun rises in Seoul there is the team which built an in-memory database as a prototype called P-time. We acquired them six, seven years ago. Then comes Shanghai as a development lab. Then comes Bangalore, a larger development center. Then comes Berlin where we have our core engineers of Max DB. Then comes Walldorf where there is the design crew of TREX, the text search engine. Then a little bit in Paris and then comes Palo Alto, and a little bit in Vancouver. And they were all on the Internet. And they all tried to know each other and because now we share pictures and everything they are a community. They vote on what their favorite food is. The last time I checked it was Thai food. They grew up with the Internet, these people. They like to share and work together. Now, 20 years later it's possible to do it 24 by 7.

Hollar: And you are, that's how you're running this project?

Plattner: Yeah, and we are. And this is the fun part. You know some have to work hard, but I have to do the opposite. I have to tell them, you have to stop. You're not allowed to come this Sunday, again. I saw you were working on Saturday. You cannot. You get addicted. "Yeah, but we have to." "No you don't have to. Delegate it. There are enough other people in a different time zone and they're not done yet. Or the Monday is already coming up and so they are fresh. So you have to take your rest."

You know that the neuroscientists are telling us that we have to relax. When we have fed our brain with information and we want to get something out of it, we have to relax. We have to go walking. We have to go swimming. There are other things, short intermissions, during which all of a sudden we have an epiphany. We have to do this. And in this environment, in this set up, you can do this. They have to do it. Despite that fact that it's hard, I never had to motivate them to work harder. I only had to tell them don't overwork yourself. Take your time off. Take your family, go to Italy. Go somewhere. Visit the guys in Korea, in Seoul but do something culturally there that is not just work, work, work.

And because we have enough other people working, they do this. They share testing. They hand problems over. We go back to 20 years before when this one American long-haired Californian dude was working on the Internet and asked colleagues in California about certain UNIX qualities and how you program this and what the experience is. And some was bug removal of operating systems or protocols. This leads to flat hierarchies. It leads to a global community. We are social people. We feel good in communities. This is the success of Facebook. Why is Facebook so much better than e-mail with all of the attributes around e-mail, point to point? Even with the distribution list, no, a community is something else. You feel safe. You feel encouraged. They might have also their problems I don't see those. They might fight. But even fight is part of our social behavior. Tremendous success. Tremendous fun.

The development team is cooperating with I don't even know how many universities. My university is part of it but they do it with others too. And everybody says the same thing. It's fun to work with you. Actually, an interesting thought comes up. Three years ago when we had a hiccup in SAP – we decided not to

continue with the incumbent CEO. I said to the employees, “I want to bring back fun to the company so the employees of SAP should have fun, again, working for SAP.“ And the German press completely misinterpreted my ambition that SAP should become a company where people have fun again. Fun is the English word for hey we have a good time. We like to work here. Just recently there was a headline: Facebookers are having fun. So people of Facebook and many other companies are having fun here.

Hollar: And what was the misinterpretation?

Plattner: This is more a cultural issue and perhaps a language interpretation issue as well. There are important differences between the meaning of luck, fun and happiness. I can say I am very happy with my partner in life. But being happy at work means something completely different. It means being motivated, energized, creating something new and working with a meaning and vision in mind that you can change the game again and perhaps even the world. So when I said SAP should be a happy company again where people have fun working - I meant exactly that.

Hollar: Kind of contentment.

Plattner: It’s more personal. Fun is a group thing. And we don’t have the right word in German for that. So it’s more than having fun as a group. And it is more than being happy. There’s not an exact English word for this either. I didn’t say fun-- I said I want SAP to be a happy company again. And the translation was not correct. When you come to SAP in the morning and you have problems and there are things to solve. And there is a hassle, there’s always a hassle but you are in a group and you can share your ideas, work together on the challenges and achieve something good and meaningful. And then you walk home and you say, all right, it was a good day today.

Hollar: Now, you’ve talked about <u>what</u> you came back to do, which is HANA, and <u>when</u> you came back, around 2007. But are you now telling me <u>why</u>? Why did you come back after you had stepped back a bit?

Plattner: I always said I will still keep an eye on development – even though I’m only chairman of SAP. And I gave myself the title “the chief software advisor,” and that’s even written, and I have a budget for that and everything, a little bit mimicking Bill Gates. Not so easy-- where should I advise them? But shortly after I stopped being CEO of SAP I started my professorship at HPI and working with the young students and that changed my perspective. I relearned IT and being surrounded by all of these young computer scientists, you have to cope with them. I have to read their papers. This rejuvenated me in a colossal way. And that enabled me to go back to SAP and to even younger people at SAP and be on par. And yes, I can probably tell you what we should do and help them to make decisions and take the leadership to build this HANA product. It was clear two-and-a-half years ago that this will be a tremendous success. We only have to pull it off.

Hollar: You talked earlier about timing. You don't want to be too early. You can't be too late. How do you feel about this now?

Plattner: It is fantastic. We are testing, already, the next Intel version, the Sandy Bridge, another 30 percent faster, another 80 percent less power consumption so we can pack even more on one board. So there's a new Moore's Law that started now with parallel processing. We figured out how we can run massively parallel. Next week or the following week will be a very important day for SAP. We get our first 4,000 core processor installed. So then we will have this as a HANA development and demo machine where we run 4,000 cores and I think 100 terabyte of DRAM. And this is only the beginning. Google has shown that they did this years ago when they took the two processor PC for $300 from manufacturers somewhere in the world and put hundreds of thousands of those in large factories and connected them and ran Google search on that or Google Earth on that. This happened in the last 10 years. So now we can compact this and use that in enterprise systems.

Hollar: You seem incredibly energized by all of this.

Plattner: This is exciting again. This is really exciting again. There's no reason to be pessimistic about the future. People are asking me-- why do you want it to be faster? Because with faster, the user experience gets so much better. We can empower users to do things they can't do now.

I'll give you an example, there's a Chinese company manufacturing white label products, refrigerators, broilers, and the normal kitchenware. And the chairman of the company has a business proposal to turn the structure of the company upside down. Normally, a diagram of a company structure is a hierarchy with a chairman on top, then the executive board and then the executives, and down here are the workers in different shades. And he turned this around and said the company should work differently and be organized differently. The most important ones are those who are in direct contact with customers -- selling services, consulting, etc. This class is an extraordinary class of people. And the rest of the company, all of the other ones have to serve them to do the optimum job. Everybody was intrigued by this. It's a very concise presentation and he's very good at that. He's a member of the party. And now we are close to defining how this can be done.

Let us assume we have a database where all enterprise data is stored. And just for the sake of argument, assume that any question you have requiring any combination of data could be answered within one second. Then go through all of the information that could be known and used by a person who is working in direct contact with a customer. That's a lot. A lot of this we don't allow them to use. It's not their business. But let's look at a potential scenario from my favorite dunning and collection process. The sales rep is with a customer and looks at his iPhone five minutes before and knows that the customer still has to pay $2 million for the products that were shipped last month. The rep nicely asks about receiving payment for the products we shipped. The customer says yes, but there are still open issues, escalation or legal reviews pending. The rep immediately sees that escalations and reclamations were all solved 23

days ago. The last one, actually, 22 days ago. Result: An embarrassed customer promising to wire the money tomorrow. The person in the front negotiating the next order-- this is the person who must be able to make the decision now. This is the person who can put the pressure on or offer to fix a problem in the next batch.

Think about a retail company and a consumer product goods company discussing with the store what should happen. Empower the people who have the customer contact to make the decision based on all of the information the company has and we can do this now. We don't need the monthly reporting system to spit out who is late and which country is late and you know that countries have different payment behaviors. I learned in China, a lot of products are being sold using cash, cash for product or product for cash. There is no credit card. There is no invoice, you pay cash, get the product, and okay. Thank you very much, next one please. Very good business, very simple. Nothing happens in the computer system. With HANA we can empower people in a way which is really completely different. This is a revolution. We can make our products so much better. We can get instant feedback from the customer with regards to the product. The whole chain of people who are responsible for the product can look at it and see how it has to be improved or changed. We can build better products using the same amount of people energy we put into the economy.

Hollar: And are you including this instant feedback in your development cycle?

Plattner: Yes. We talked earlier about the new little applications called "recall" which was developed by people in the SAP Palo Alto app house. And I asked the developer whether he had customer feedback and how he handled it. And he didn't answer. When I dug deeper we learned that feedback was already available through the government recall registry. Somebody has to know how to go to the right computer and the feedback may be harder to pursue and less elegantly presented. So we didn't invent the information. We only present it in a different way. We take it from this central server, a U.S. government server into the HANA database and there is extremely fast access. So I believe we will transfer speed into empowered behavior by people inside and between companies.

The faster Google became on the Internet, the more I became addicted. The Google Internet search is speed. Without the speed, I wouldn't use it. Since it's the fastest way to find something I use it. So I believe there's speed and there's a little bit more than speed. We can do structured data and unstructured data in one system now. We can analyze text, index text and then run SQL logic on top of both. In memory storage basically everybody who builds an in-memory database builds it on the principle of columnar store which was invented 20 years ago or even earlier. But this too has disadvantages when your primary storage medium is disk and DRAM is only cached.

Hollar: I was talking to one of your engineers outside earlier and he was talking about this, columnar data.

Plattner: Yeah, we can basically -- any single attribute we have is an index in the system. And we scan with an unbelievable speed of two megabyte per millisecond per core. So if I have 4,000 cores you can do a calculation to the megabyte per millisecond per core. But what amount of data can we scan in a second? One second is for me a critical timeframe within which we have to come up with an answer, so we don't lose the attention span of people. And you can only empower salespeople if it is instant. If it's not instantaneous, they don't do it or delegate it to a sales assistant or somebody in the back office and we are right back where we started.

Earlier I talked about the doctors and clinics that we do research with, in the 60 mile zone around Berlin. For the first time, we get the doctors on the system because a doctor has no time - five minutes for a patient. And they have to permanently switch gears, get the new facts in, look at it, and make a decision. You cannot work with them when they have to wait 15 minutes. But what if they can identify in a second a type of illness, and potential therapies and a prognosis and do it on an ipad or smart phone at their fingertips. This is pretty clear, we should do this here. He or she will use that information. And it is not only they who use it -- the whole world can benefit from this acceleration. One doctor starts thinking about it and gives the rest feedback.

Hollar: But if he can do it with something that's in his hand instantaneously---

Plattner: Yes, instantly he or she says can you do this for me as well? And if we can do this probably the next day, we keep them engaged. So keep people engaged. We are so much better when we stay in the mood, in the zone, when we are part of a team. This comes into play with some of the exceptional insights in design thinking. A lot of the training that they do at Stanford is to have very short, very brief sessions, make up your mind now, it doesn't get so much better after an hour or 24 hours. Yes, there is potential but let us work for a very short period and we already have 80 percent, probably already 90 percent of the potential. Then let multiple brains start working on it and push it around and build a prototype to let the hands start "thinking" because when we do something with the hands, all of a sudden a different part of the brain is engaged. We look at its spatial qualities -- and very interesting things happen when people start actively using a concept or a mockup of a product. So therefore, I'm very optimistic that we will build better and better products with the same amount of energy, which is good.

Hollar: Let me ask you-- I want to be mindful of your time and I just want to finish up with…

Plattner: You probably don't have enough tape anymore.

Hollar: It's all memory now.

Plattner: It changed, but the cameras are still of the same size.

Hollar: Yes.

Plattner: I heard you need this weight on the shoulder

Hollar: That's right. Well, videographers get insecure if they don't have the large camera to work with.

Plattner: My daughter and her boyfriend are both actors. And I gave him a Canon still photo camera which can do HDTV-- HD movies. And they are shooting a movie now and they do the pre shots of the movie with a Sony camera. There you go. HDTV, it's unbelievable.

Hollar: It is unbelievable. Let me finish with some thoughts on leadership. You've already started to talk a bit about how what you've experienced with technology and changes in going back to university and being with young people has influenced you. What would you say, as you look on your career now, you've learned about leadership that you're actively applying today in this new project?

Plattner: Yeah, there's one thing but you can't turn time back. There are so many things I regret and with Monday morning quarterbacking you could have done better, you should have done better. Why didn't you know that? That's the interesting part. How can we ignore facts which are there, clearly visible, but still we ignore them.

Look how many companies didn't see it. The tragedy of Ken Olson, CEO of Digital Equipment Corporation, who was ignoring the PC that in his mind could never work. He was convinced that he had a much better operating system, everything better. And he lost. And a lot of his people went to Microsoft and created Windows. The tremendous success of Windows, why didn't he see it? Or you see it and you ignore it because it is not comfortable.

So one thing I learned, you have to face the complexities of life, of development, of a company and you have to try to solve them in an intellectual manner. You have to discuss them. You have to be open about them. You cannot hide them. You cannot push a button and force them. You cannot order them. Yes, you can push people. You can tell them come back with the report in one week. You can do things like this. But you cannot really squeeze innovation out of people. You have to enable them to do innovation. So the key thing about leadership is to enable people to be creative and get things done that are necessary. Create an atmosphere of openness, this is good management, to encourage and promote a different style of collaboration.

I mentioned the "SAP app house" as an experiment. I mentioned design thinking which has very, very good traction now. Things are not easy. We have to face problems and we have to try different means to solve problems. When we talk about products, we always have to balance our customers, the product,

and how our customers see the product and our own business. This is probably the most important thing because it means that we have to keep the customer happy. They have to like us and like the product. We have to keep our developers happy so that they build something which is feasible and don't stop too early. And we have to think about how we make money. Without money, everybody learns that in the MBA class or earlier, we need a business plan. We need an idea about how we make money. I see this, for example, as a little bit harder for the German IT folks building companies. And the Americans have a more natural relationship to money and money-making. And this is one reason why they are more successful in this business. And the Chinese are even more money-oriented.

Hollar: Isn't that interesting?

Plattner: This is really interesting that the Communists are beating us. I cannot stop laughing about it. But everybody knew before-- wherever a Chinese family came into the world, they became a center of success. So actually capitalism is their natural attitude, probably. Yeah, so it sounds probably too philosophical but use your brain to solve problems. That's why we have it. We can't solve so many things and so often we make mistakes. We ignore information. We ignore people because they tell us the wrong thing and we don't like what they say. Yes, this is not good.

But when do you incorporate this and when do you make a decision to change course? That's hard. You cannot permanently change course. If you do it too early, you fail. The famous historical example is Newton, Apple had all of the right ideas, just ten years too early. Then still at the right point in time the iPhone pops up. You still can't write but you can do nearly everything else. You can talk to it now, it's even better. And then once you have made a decision then put the pedal down and try to accelerate. You see the innovator's dilemma is happening again and again since that book came out.

For large companies you have to split your mind and do-- at one point in time you have to do multiple things in parallel. So for SAP now the decision is yes, we do on application and we do on demand applications in the cloud. We do both. And we have to live with the complexities in sales, in development, in management, and everything. We have to live with this because that will be the world for the next ten years.

Hollar: Do you see yourself doing this as long as you're happy doing what you're doing?

Plattner: That's a question I don't really like but I can answer it. At the moment, I have a lot of fun. I'm up for re-election on the supervisory board of SAP. Whether I can do this for another five years with the same intensity I don't know. But it is so much easier to do what I do now than to be in this corporate straitjacket of quarterly reporting, and you have to do this and that and you have to make all of the other management decisions, and decisions for your employees. And you're permanently under time pressure. Today, I have no time pressure-- except if I don't have an idea today, probably tomorrow-- or I just talked

to somebody who could have an idea. So this is-- yeah it is much more on the fun side. And when you see the success of your engagement then it makes you really happy. So this is instant reward or pretty nice reward coming and so it's a good time.

Hollar: And so since years ending in two seem to figure so heavily maybe in 2022 you and I will sit down again and look back on this coming decade.

Plattner: I was born in '44, so that probably means a philosophical résumé about the then 60 years in the digital world. But what a revolution, what a dramatic change, what opportunities from carrying around punch cards 40 years ago.

Hollar: That's right. It's been a pleasure.

Plattner: Yeah, thank you. It was a good time.

Hollar: This is very helpful for us. It has really helped us. And we're building an archive-- we're approaching 600 oral histories, now. And we're building an archive that I hope will be as much about business history in ten years as it has been technical history because researchers want to know about these…

Plattner: And you have all access to the histories of Andrew Grove and Gordon Moore?

Hollar: That's right. We've taken their oral histories. And now, we're starting with some of the earliest venture investors. And then we're going to begin to…

Plattner: Vinod Kosla and those ones.

Hollar: That's right. Bill Draper. Tom Perkins. And in about three weeks I'm going to do a team oral history of Steve Wozniak and Mike Markkula. So we'll talk a little bit about the beginning days of Apple.

Plattner: Yeah, he came once to SAP and gave a speech and you see the similarities there. They had fun building things. So they didn't see the whole future. So you stumble into it. You see a little success. And therefore, I believe this new world where the instant feedback in the Internet will stimulate innovation and sort out lack of innovation.

If you can't find the user-- he became a board member now. Lars Dalgaard from SuccessFactors. He is a board member now of SAP. And he runs the whole cloud business of SAP. And he came to a supervisory board meeting and just remarked about things we do and said, "this is just not relevant." We were sitting there. Why do you say not relevant? We spent over a billion dollars. He said, "It doesn't matter how much you spend. If you have no users it's not relevant." So this is voting with the user key, the username. It's pretty brutal. Therefore, no Facebook page, this horrendous amount of money for something they can use or probably they made the right decision. It was the users and the growth rate they were enticed with. Not the money, not the team size or whatever. No. It's if you have critical mass and momentum that is behind this growth rate – this is what really matters.

END OF INTERVIEW

