Oral History of Joel Birnbaum
Part Two

Interviewed by:
Chuck House

Recorded: March 19, 2014
Mountain View, California

CHM Reference number: X7119.2014

Chuck House: OK. Joel, it's good to have you back. It's Wednesday after our last Friday interview, and we've got a few more hours to go, right?

Joel Birnbaum: Right.

House: Well, pleased to have you. And last time we covered a lot about IBM and didn't get too far into the HP story. So, I think today we'll concentrate on HP.

But before we do that, let me just finish up a few things with IBM. This thing about the museum here-- we have 64 fellows that are acknowledged in our Hall of Fellows Fame. Nine of them are from IBM. None from HP. So, that's kind of an interesting disparity right there.

And I suspect you know every one of the IBM players-- Backus, and Eric Bloch, and Bob Evans and John Cocke, obviously, and Fran Allen. So, let me just go down that list. A couple I hadn't thought about-- Don Chamberlain, you probably knew Don? Probably knew him pretty well.

Birnbaum: Pretty well. He was at San Jose.

House: Al Shugart was there at that time, probably in a different group.

Birnbaum: I didn't know him.

House: And Gene Amdahl, of course. And Fred Brooks I think probably had left by the time you got there? Maybe right around then?

Birnbaum: But he was a consultant, and, in fact, he was a consultant that I hired and was a major input to the RISC machine development [INAUDIBLE].

House: Any of those that you want to comment on your interaction with them and help they might've given you and vice versa?

Birnbaum: When I became the head of the computer sciences activities at Yorktown Research in 1975, I inherited the IBM Fellows who were attached to computer science. And, at the time, I think there were about 40 IBM Fellows worldwide. After quite a number of years of it, the program was, at the time, a lifetime appointment and very prestigious. And I think something like seven, or maybe it was nine, actually reported to me. So I actually was the manager of a few of the people whose names you mentioned.

House: You mentioned Goldstein last time.

Birnbaum: Herman Goldstein, yeah. Ken Iverson, who did APL, and of course John Cocke. A Fellow named Andy Heller, who had done all sorts of system things but had gotten to be a Fellow mostly for doing VTAM, which was an access method for virtual telecommunications, and a number of others.

They were really fine people. I think any one of them would be a kind of person you would think of as holding a chair at a major research university. And they kind of functioned that way. Many others-- Benny Mandelbrot on fractals. And these were really giant people. And for a while, I was working jointly with Shmuel Winograd, the Winograd Fourier transform, and many other things.

House: Terry his son by chance?

Birnbaum: Pardon me?

House: Is Terry Winograd over at Stanford related to him?

Birnbaum: No.

House: Not at all.

Birnbaum: No. Shmuel's Israeli, and I've asked each of them separately that question. They're not. Shmuel's a mathematician, and when I was the head of the computer science, he was actually the head of mathematics. And a number of others, all very distinguished.

They were, as a group, something of-- they comprised a significant part of the Wild Duck population of IBM research. They came and went as they pleased. They had high salaries, as if they have become senior managers. They had budgets that were variable, depending upon what they were doing. In some cases, it was just a salary and a generous travel budget. In other cases, they had people assigned to them, did projects, and so forth.

I had a very unusual IBM fellow, perhaps one of the smartest of all of them, work for me for two or three years. His name was Ian Gunn, and he passed away not too long ago. But he was, I think, one of the world's great physicists, even though he'd been trained as a mechanical engineer at Cambridge. He had only a bachelor's degree, but he seemed to have the equivalent of a Ph.D. In anything you chose to talk about.

House: He wouldn't be the Gunn diode?

Birnbaum: He's the Gunn diode. He also, as a graduate student, had designed the suspension for the Lotus with Colin Chapman. And he was the oldest man, at the age of 52, to ever win the United States motorcycle motocross championship. He was about 14 years older than anybody else.

So he, of course, was in physics department, but he really fell in love with computing. And, in particular, he fell in love with APL, Ken Iverson's thing. And he asked whether he could switch his IBM fellowship to my department, which you could imagine the IBM management wasn't too happy about taking one of the really major physics contributors. And he wanted to write an editor for APL.

It turned out that-- so, of course they said yes. And he spent a couple years on it. While he wasn't bad, he wasn't particularly good, either. And gradually we nudged him back into the physics department.

My principal job was described to me by my then boss, another IBM Fellow Ralph Gomory, who was a mathematician and former head of the math department and the head of all research, and he said, you know, you sort of don't have to worry too much about what Ian does, but you do have to worry about keeping him alive.

So, if you would not mind, he drives a very, very fast sports car. When he's not driving his racing motorcycle to work-- and he races that motorcycle in these crazy dirt-bike kind of events, and it scares the hell out of us. He's broken many of the bones in his body-- so would you please do what you could to see if you could dissuade him from any of those activities?

Ian and I became a quite good friends, but I have to assure you that the closest I ever got was he allowed me to come and root for him at the events, rather than ever stopping him.

House: You didn't dissuade him.

Birnbaum: Yeah. So we had many, many interesting people. Rolf Landauer was one of the Fellows, but he was also the Associate Director. But he spent most of his time in our department.

He and a guy named Charles Bennett, who also later became a Fellow, were laying the groundwork for what they call program dynamics, in which they showed that you could analyze workflows and working sets using a lot of the tools for theoretical physics and a lot of those approximations.

Sachi Kobayashi was there. He wasn't yet a Fellow-- he became one later. He ran the performance analysis, but he was a world- class mathematician slash performance analyst. And he invented something called the diffusion algorithm, which was a way of figuring out the behavior of caches and other things and complexes.

House: Did Dominic Ferrari work with him at all?

Birnbaum: Pardon me?

House: Dominic Ferrari work with him?

Birnbaum: Yes, yes.

House: I knew him at Berkeley.

Birnbaum: Maybe. And Don Fraser. Yeah, well, Kobayashi went to Princeton. He was the founding head of the IBM Japan Science Center, which I was involved in setting up, but it started after I left IBM. And I had nominated him, and he was elected to that. He was a wonderful director. I don't know if he ever went back to IBM, but he became the dean of engineering at Princeton and professor there.

House: Jim McGroddy? McGroddy?

Birnbaum: He was in charge of the technology department when I was in charge of the-- well, for a while, he was in charge of the technology department. A guy named Holly Caswell was my exact contemporary. And he was made the head of technology at Poughkeepsie, which was a large IBM site, of course. And Jim McGroddy was the guy who came and took over technology, which was distinct from physics. He drew a big distinction between the two, although they often collaborated on projects.

House: Then you talked about Eric Bloch a little bit last time and Ralph, of course, and Lew Branscomb. Those people all cycled down through--

Birnbaum: Lew Branscomb came later.

House: Oh, did he? OK.

Birnbaum: And he had been the head of the National Bureau of Standards. And IBM decided, for some reason-- probably one of its board members had suggested it-- that we needed a chief scientist in addition to a director of research and in addition to a guy who headed a staff that did engineering, programming, technology oversight for the whole corporation. That became Bob Evans later in his life.

And so we had a chief scientist, and everybody kind of liked Lew and respected him, but they couldn't understand what his job was. And I think he didn't know what it was either, because IBM had never had one. And he kind of tried to invent the job, which essentially was a staff review of most of the projects and research and elsewhere, and making sure that research was coordinated. And it turned out that of course Ralph Gomory thought that was his job. And so there was a--

House: A little contention there--

House: --More sound than light that came out of that relationship. But he was very, very interested in what we were doing and was a big supporter. He came by and was a very, very quick study. Even though he wasn't a computer scientist himself, we found he could understand everything very quickly. He seemed to have a very good sense of which projects were likely to pay off and which ones weren't.

And, from my point of view, not being the guy that had to deal with him on a daily basis, he was a friend in the corporate headquarters. He could, on an almost daily basis, help to explain to the executives-- who were mostly people who came from marketing, many of whom had no technical backgrounds at all-- and he would help to put things in terms that they could understand. And he would, in general, be very supportive of what was going on in research division.

And in those days IBM had a-- I don't know if they still do-- but in addition to the development laboratories-- of which there were many which built products and systems-- and the research laboratories-- which in those days there were two, San Jose and Yorktown-- and Zurich, rather. There were three. Then they added Japan and some more later-- there was also a third layer, and it was called Advanced Technology.

And these were people that were supposedly in between research and development. So they did things that were a little too risky, little too far out for the development organizations, who were mostly focused on manufacturing, costs, serviceability, schedule, and so forth, and not far enough out, not risky enough probably for most of the things we did at research.

It was a very uneasy partnership. You didn't know whether you were supposed to transfer it to them and then to the divisions. And since we had a number of failures doing it that way, we tended to bypass them. They, on the other hand, often found that they needed to do research in a particular area, and so they didn't mind doing some of that themselves or contracting with a university.

And it went on that way for most of the time I was at IBM, largely ignoring each other. That was pretty hard to do, since their headquarters was about four miles up the road in a place called Mohansic, New York, which is where we were. But I think some good things came out of that, but compared to what came out of research it was nothing unusual.

House: So that's a great prelude now to move to HP, because not only they're 14th, or maybe 25th in computing in the world. Not even a postscript dot in IBM's view. And the cultures were somewhat different. You covered that last time. Those kinds of organizations didn't exist at HP. It was a very different structure.

Birnbaum: When I came to HP, it was a really different structure, because there was the HP Labs, and there was the company. And the company was manufacturing and marketing and R&D, in a nice little triad, each with a manager, with the general manager over them. Every tub on its own bottom.

When the division got bigger than about 1,000 people, or about $100 million or so in those days, it would tend to split and form a new one. And usually one of the triad members would become manager of the new one. And so there was a nice mechanism incenting people to do it. Then people had a lot of local control, almost complete local control.

As the company got bigger, the divisions got organized into groups, but nevertheless, the distinction between what was researched and what was not researched, let alone this complicated advanced technology thing, it just didn't exist. They were partners.

There was no separate set of titles. Everybody at the research was called research staff member in one way or another. And the people could go from there to work in the division. They often did to help transition the technology to manufacturing.

Sometimes the people from marketing or from manufacturing would come to the research organization. You didn't get a raise. People didn't worry about whether you could go back or not, because if you were a good person, there was going to be a job for you somewhere. Maybe not the same job.

And that was, to me, just the most wonderful thing, because here were these guys that at IBM I practically had to stuff things down their throat to get anything to happen. Usually it resolved-- it would escalate up to some high level management team would decide whether the research prototype was something that was going to become a product.

More often than not, they decided it wouldn't, because IBM felt that they didn't need many of these things or that they were too risky. And here in Hewlett Packard, they expected that a reasonable fraction of the most advanced things would come out of the labs. After all, wasn't that what the labs were for?

And that, of course, came I think from the personalities of probably many people, but certainly Bill (Hewlett) and Dave (Packard), and also Barney Oliver. And these were guys who could hold their own with the development people. They understood about what it took to make something real. At the same time, they had enough credibility with all the academic and research types that they could certainly mix it in with the best of those as well.

House: With all due respect, those labs were, as you said, to serve “every tub on its own bottom.” The division labs were small? They were constrained almost by the size of the division itself.

Birnbaum: Yeah, they were very small. They had a little rule of thumb formula as to what percent of R&D that-- R&D in those days was a high percent, something like 10%, often as high as 12% of revenue-- and the labs would typically be about 10% of that. And so when I came, I think HP Labswas well under 500 people, counting the support staff and the technicians and secretaries and everybody. So there probably were, I don't know, 350 or so I'll call them researchers.

House: But when I came out to Corporate Engineering, which was almost the same time you came to HP, I had 91 divisions. 91 R&D managers sort of quote reporting to me, and none of them were big. They all had their own fiefdom. And so the thought that they could do any research was crazy. So the labs was a natural feeder.

Birnbaum: And it was mostly a very friendly, very nice relationship. It wasn't so easy to transfer things, because usually they wouldn't wait for the labs to finish their longer-term work, and they would have their own things going on. After I got the lay of the land, we developed the feeling it was quite a bit different than what they had done before. When I came, most of what was going on was applied research, with a definite intent of transferring something for something that already existed. There wasn't too much that blazed new ground. Every once in awhile, something would come along like the calculator, which would. But, in general, most of the things made the products that we already had better or created the next generation of those products, or, in some cases-- few-- disrupted the product and produced something better. Time domain reflectometer, or something like that.

House: So against that backdrop now, the story you started to tell last time about Spectrum and the difficulty of getting that inserted on the body politic is kind of an irony.

Birnbaum: Yeah, I guess I must've really been nuts. I don't know why I came, frankly, but everything was kind of stacked against success. Maybe that's what I liked about it. The first problem was there was already an in-place effort. And it was quite large and it had very good people running it and people who had long- time, well established, well-deserved HP reputations as guys who solved problems and got things done. And they were hard at work at that.

So, the second thing was that I didn't believe in what they were doing. I was pretty sure it wouldn't work, because it was very similar to something that IBM had tried but never published, except it were smaller and, I thought, quite a bit more primitive. And it lacked the one component that might have made it successful, which was the optimizing compilers. They didn't even know what an optimizing compiler was at HP. I mean, they literally didn't know. I used to have to tell people what that meant. And what it was that you optimize, which was quite astonishing to me.

And the third thing that was wrong, was HP was used to promoting from within. A great many of the people that had been successful, I'd say most of them had come to HP quite early in their careers. In some cases, maybe many cases, directly out of school. And they'd spent a good portion of their life in a very paternalistic company that treated them very well, and they got promoted on the time frame that was in.

And suddenly here comes this New York character from IBM, no less. They parachute him in and they make him instantly a director of a center. And he's the one that's going to lead them into the promised land of computing, even though they've already got three fairly successful computer businesses.

So the successful-- meaning that they were selling a lot. They weren't making a lot of money. And their future was cloudy, and they clearly didn't have enough resources to support all of it.

So the next reason why I shouldn't have gone to HP was that what they were asking for was to sort of come up with a combination of the three architectures in one new one. That's what the people in the divisions were doing also. That had been attempted, in some sense succeeded, with the System/360 and a few others, but it required a lot of new application development and a lot of new compile work and so forth.

That would have been a really, really hard challenge, except there was a small caveat put on which was that, by the way, we want you to be the best cost-performance computer in the world, bar none, in addition. So they wanted the best computer, but it had this gigantic constraint, this albatross around its neck, that it had to be backwards compatible with not only the other architectures, but with all the applications that ran on the other architectures.

Birnbaum: As it happened, I had spent a lot of time thinking about that at IBM with the risk machine development. And I had the feeling that, if I could find guys that knew what the people at IBM-- if I could get people to the point where they could do those same kinds of things, that we had a shot at it.

That isn't the only thing they wanted me to do. They wanted me to get them into a number of other computer-related businesses, like databases, that they were just tinkering with. And, of course, to eventually maintain the important role that all the instrument- related divisions had had, because a great many of HP's instrument lines either counted on the labs for help or, in fact, in some cases had actually been founded by things that were invented at the labs.

So, it seemed a tall order. And when I look back on it now I wonder why in the world I would have done it. But I had become very disillusioned at IBM by not having things transfer.

And I felt that, if I was really ever going to amount to anything as a manager, which was the path I had chosen, rather than being a technologist all my life, that this was the right size kind of challenge. Kids just gone to college, so the two of us were free to move. California always seemed a wonderful place to live, so we figured, what the hell, we'll give it a try.

It was real hard. And what was hard was the technology. That was really a big challenge, but we had good people, and they tried a lot of interesting things. And some things they came to were the same conclusions at IBM. A good many of them they came to their own conclusions and did them differently, and in some cases, better.

So we solved the problem of how to make a fast machine, that was backward compatible. We developed the migration centers, which we talked about, that would bring the customers across. But the thing I hadn't figured out, and it came back to bite me very badly in a later project because I never figured it out, was how to get across the political chasm.

Because here were all these people who had worked just as hard as we had worked, and they believed just as much in what they were doing, even though they weren't having the kind of success we were having. Their performance was not reaching its goals, their cost targets were expanding, the number of people they needed was getting larger. But they weren't quitters, and they had solved many problems before where the going hadn't been smooth.

And there was this notion-- I ran into it at HP's board of directors and more than twice, I think, at the executive committee, somebody like Barney Oliver or one of his contemporaries would say-- "Oh, for Christ's sakes, some of this software's taken up too much space. Let's just take all these guys and lock them up in a barn for two weeks, and they don't get to go out until they've shrunk it in half," like it was a power supply that was using up too much current or something.

They had this notion that you were working on a product, and you could bring all the brain power you could get, not all of which had to work for the company-- hire consultants, bring in professors, form joint projects-- just solve that problem.

And I would patiently, over and over again, try to explain to them that I'm sorry, but you want to compete against DEC and IBM, or all the rest of them, those are systems companies. And when IBM brings out a computer, there's something like 40 different divisions that are involved before they're finished. They merge them and test them and validate them. You can't lock ten guys in a room. You could solve the individual problems that way, but you can't bring out the system that way.

And I remember painful lectures that I have to give to people, and I could tell the whole time that I was talking that they probably didn't believe me. I never had heard the word sandbagging before, which means essentially to make something look worse than it really is so that you'll look better when you're successful. And I'd never been accused of it, certainly, but boy did I get accused of it then.

And I kept telling them you can't just take that operating system and throw it in here. It's not just that the instruction set is different. The IO is different, the cache is different, everything is different.

House: You got there about the time of that N/MOS program out of Fort Collins, right?

Birnbaum: Oh yeah, it was finished.

House: Which was exactly that kind of problem.

Birnbaum: It had just finished.

House: The chips were wonderful--

Birnbaum: They just made a 32 bit processor--

House: But there wasn't anything else.

Birnbaum: And in fact, during my entry exam from Barney Oliver, he'd shown that board. I'd been impressed that they'd done 32 bits and had really what looked like a very fast technology.

And I remember I asked him two questions, and he was so annoyed that I had asked both questions. I said, "What are you going to do with the heat when it gets fast?" And he said, "What do you mean, what am I going to do with the heat? Well, blow fans on it." And I said, "No, you won't."

And I said, "Whether you like it or not, you're going to be in the thermal grease game and you're going to be in the pump water game. And I don't know what kind of portable computer you think you're making here, but when this stuff goes as fast as the [INAUDIBLE] is going to go, you're going to be out of it. You're not going to be able to." Well, he didn't like that at all and didn't agree with it. Neither did Fort Collins. Turned out to be true.

And the other thing I said is, "What operating system are you going to run?" And he says, "Well, we have our own. It's called Rocky Mountain Basic." And I said, "Rocky Mountain Basic? What in the world is that?" I said, "Isn't that a language?" He says, "Well, it's a language embedded in an operating system. It's really slick. Wait till you see it. And I said, "Well, excuse me, how's that going to do online transaction processing? Are you going to run COBOL on that?"

And so on. So that was the nature of my early exposure to that. And I wound up giving lectures to these guys. I mean, they weren't really lectures, just discussions, and I'd say, you know, one of the things I learned, I learned it from Fred Brooks, in fact, early on in my life at IBM, and I heard him talk about it, that you can't just do a computer design where you focus on one factor.

You have to have at least three things in your mind. And, in particular, you have to worry about functionality, performance, and what he called manufacturability. What they used to call RAS-- reliability, availability, serviceability, and so on. And all of that within a core structure that's competitive in the market.

And if you only worried about putting, for example, as much function in as you could, which is what division people were doing, well, then the performance was going to suffer. And if all you did was optimize the performance, almost certainly the security would pay a price, because you'd be putting in high speed paths everywhere that weren't adequately protected, and so on.

So you needed a design team that knew about all those three things, which is why I always had any team I was ever responsible for, I always had all the people physically sit near each other. Because the difference between having to walk across the street and walk across an aisle when you're trying to decide something about which is software, which is hardware, and which is built in, and which is microcode, and so on.

If you could just do that on an hourly basis with a simple question with someone that was your friend, as opposed to having a meeting and reach a decision and start a task force, which was the IBM way because it was so big and it was spread out.

So, it was a real learning process for HP. I remember that I walked into the buildings, and they have these desks just out in the open. There weren't even partitions at all.

And I said, "I'm sorry. It's not enough to have a conference room for people to get together. People are really thinking deeply here, and they can't be constantly distracted by people walking back and forth, back and forth. I don't care if they have so much acoustic privacy. I want them to have visual partitions."

"We have partitions." I said, "No, they're the same height as the desk. I want them to be high." So, the management at the time, John Young and others, Paul Ely, all of whom were sitting in building three with the low partitions, said, "I don't think we can do that. We've existed this way for a long time, and we've done it a long time."

So I said, "Well, I don't think I could recruit any of the half dozen senior people that I think I need to pull this off. Because I've had a few of them come in and they've looked and said, "Are you kidding me? I can't possibly work in this."

House: It looks like a Lockheed Martin bulkhead.

Birnbaum: So, it worked real well when you had manufacturing and development and everything all together, but it's not good for research. Or I didn't think it was, anyway.

So I went to see Bill Hewlett, and he called in Dave Packard, and we sat down. And I said, "I really want to put in some partitions here. And we're moving into building three from the little temporary building we're in. So, when we move, I want to have these things, and they're going to look nice."

And I showed them-- I had a guy come in and give them a thing. You could see they were just aghast. They thought it was horrible. But I think I made a very good case for why it would be helpful, at least for recruiting and probably for working. And they said, "OK."

And about a week later, I got a New Yorker cartoon from Dave Packard in my mail without comment. It just had his signature on it. And it was a cartoon which showed the newly evolving open offices with these high partitions. And two rats were standing there in the far side with a stopwatch, and the guy who was supposed to be a scientist in a white coat was about to run the maze of partitions. And one rat is saying to the other one, "They'll never be as good as we are." So he sent me that.

So, we got our partitions. Interestingly enough, not long afterwards Cupertino installed them as well, and then Freddy Wennnger installed them in Fort Collins. And before long, nobody could remember when they weren't there. But it was some struggle.

House: I had not heard that story, and that's a great story.

Birnbaum: It was some struggle to get them, and people realized that you could still have the openness where you could just walk in. But, gee, if you're really, really thinking for a long time, you just don't want everybody that goes by, "Hey, Chuck! How's it going today?" I mean, you just don't want that. You want to be private. Not for reasons of secrecy, but for reasons of concentration.

House: We-- I had partitions from day one in Colorado Springs. I never had them out here.

Birnbaum: Oh, well then I'm wrong. Then we weren't the first one. But we were certainly the first one in Palo Alto.

House: And it never occurred to me that that would even be an issue. My God.

Birnbaum: Oh boy, was it an issue. And then came the second issue. And the second issue was harder than the first issue. And this was a sort of over- my- dead- body- kind of issues. So one of the things that HP was really proud of in building three, the former corporate headquarters, was the shine on the vinyl on the floors. Because every night two guys came through--

House: Oh yeah, they had those big polishers--

Birnbaum: --Those great big polisher machines. And every night they polished them, they looked like mirrors. The building was, I don't know, 25 years old at the time, but the floor looked absolutely brand new.

And they pointed it out to me. In fact, even when I came for my job interview, one of the people that was walking me around, he said, "Boy, look at this floor. It looks like nobody ever walks on these. Eat your breakfast off it." I heard that expression several times.

Well, guess what. We started hiring more people. We started having more secretaries. A lot of the secretaries wore heels. Lot of the guys didn't wear, in those days, sneakers. They wore shoes. It made a lot of noise when they were marching around. A real lot of noise. The coffee wagon would come, all work would stop for 15 minutes while this thing came grinding down the hall.

So I said, "I want a carpet. I want to put carpet down. At least in the aisles, but maybe everywhere." And they said, "Look, you've got the partitions. You're not getting the carpet. This floor is--

House: Is that right?

Birnbaum: Absolutely. Totally forbidden to have the carpet. So I didn't know what to do.

House: Dave and Bill were still there?

Birnbaum: Absolutely. They were there. You bet. They were there. They're the ones that said no. John Young said no too. Good enough for 40 years with a linoleum floor, you don't need carpet. So I didn't know what to do.

So I went to an old-time HP guy and I said, "How can I do here?" He said, "The one thing these guys listen to is money." He said, "If you can make a case, which you probably can, that once you get over the first cost of installing it, it would be a lot cheaper and easier to just vacuum that rug at night than it is to polish those floors, that the thing will pay for itself in a year or two. If that turns out to be the case--

House: You're in.

Birnbaum: You're in. The other thing is, we're always moving these partitions around, because we're making new labs and so on. If the carpet can be in squares so you can move the squares around, then it could look really nice. Because you can have one square inside the thing and so on. So I said, [BLOWS KISS] "Thank you very much."

We went away. We did all the arithmetic. It did indeed turn out that in the long term-- it wasn't a couple of years, it was longer than that-- but it absolutely paid for itself and was a better work environment. I found a bunch of studies that people had done about noise in the workplace, and so on.

I didn't have the guts to go and present it. I just had the report and I sent it around. And I got back a note, I think it was from John Young and he said, "OK, we'll try it on the top floor of building three only for six months." Excuse me-- three lower only, because those guys were on three upper.

House: Those guys, not us.

Birnbaum: Underneath us. "Let's try it down below. And if it works out OK down below, we'll consider it." Well, it was such an immensely popular success. The noise level dropped immensely.

House: 30 dB.

Birnbaum: Then after I finished that fight with them, I had to have a fight about the skylights. Because the building had been built with saw- tooth type skylights, which let the light come in.

House: All facing north.

Birnbaum: Which was really nice. And it made for a lovely environment. And it had a very nice-- an extremely nice air conditioning system that, as a building got warmer from the sun coming in, it even had a set of automatically controlled shades if it got hot enough that could close it off.

But in general, they liked to keep them open. It was programmed to keep it open. So it'd be nice and light. And, indeed, it was a delightfully light thing.

Perfect for working, unless you happen to be using a screen. If you happen to have a screen, it had so much reflection you couldn't see the screen. So I said, "Skylights? I don't think this is real good for computing, because you can't see the screens? Well, isn't there some kind of a hood you can have?"

So, we had the model shop make some kind of a hood-- it probably cost more than the display did by the time we got done with them, because they were all these wonderfully skilled guys in the model shop-- and they made a whole variety of different designs of hoods to kill the glare.

None of them were really satisfactory, especially when people started using-- in those days, we often used touch pens instead of mice. And so if you had to reach up into the corner where the hood was, it was just bad.

So then we had to make a modification to the shades. And basically it was nothing that had to be done, they just had to change the program so that-- most people didn't even notice it-- but most of the time the shades were at least half closed so that the glare could be controlled.

So, there were a hundred things that nobody ever talks or ever thinks about that were different about the cultures. And there was nothing wrong with the HP culture. In fact, there was everything right about the HP culture for a group of product- oriented, independent divisions and the research organization which did applied three to five year research, by and large, that supported them.

But if now you wanted to become a systems company and have people working with each other, first of all, they hadn't any infrastructure. There was no network that connected the buildings together or that connected the things together. We had to dig up Page Mill Road and put in cable.

So that generated a whole new set of cartoons at the annual meeting about how the research had gone into the construction business, and "Gee, someday they're going to make a computer, but right now they're laying wires and trenches. We're not sure what's really going on."

House: But just think about what you've just said. You're hired to come save this company from itself in the computing industry, and you're spending time on, can you have carpet, or can you have partitions, or you can you cut the glare?

Birnbaum: Well--

House: I mean, it's just--

Birnbaum: I had plenty of help.

House: Oh, but the minutia that you're having to deal with while getting ready for the battle is just delicious.

Birnbaum: Yeah, and we didn't have a lot of time, because HP was already late in its market. The stock had already started to slip. The world was moving to 32-bit computers, HP computers were all 16-bit. In addition, the 32-bit computer which was coming was only a desktop, and it was really optimized for scientific computing in real time.

House: Well, and Rocky Mountain Basic.

Birnbaum: And Rocky Mountain Basic and a couple other strange versions of FORTRAN. You don't have equivalent statements, how come? Well, we never found anybody that needed it. OK. So, it was a tough slug. But we got it.

House: What would you go home and tell Eileen?

Birnbaum: I didn't go home very much. I was there most of the time. I was excited. I loved the people I was working with, and I liked their attitude.

And there was nothing that we couldn't do. I mean, people weren't being difficult, they just hadn't come up against it. And most of the time, if you explained it in a rational way, they said sure, go ahead. Oh, you can’t hire a guy with 20 years experience and give him two weeks of vacation? Oh, OK. Well, give him whatever he had before. But then we have to do it for the rest of the company too, so, if someone there hires in--

So, one of the things was that I got the company to pass a group of special rules, but they were so democratic and fair and high integrity that, if they gave mortgage assistance to somebody to come from the East Coast to work in the research division, then if someone else came they would give it to him.

So, I was very, very happy. I was busy-- it was frustrating-- but I had plenty of time in that beginning part to get through this stuff, because I was just building a team.

And I was evaluating the 25 or so projects I inherited with 50 people. Many people were in three or four projects at the same time. At least half of the projects suffered from what I had learned at IBM was a fatal flaw in research projects.

In IBM you physically had to say in describing your project at the annual review, is thisa science project? In other words, is it curiosity-driven after basic discovery, and what might come out of it might be an invention?

Or is it an applied project, in which you really have a customer that would love it if you could succeed. You're after innovation more than you're after invention, making something new or repurposing it. And it has a shorter time frame.

And we recognize that you need to measure those two kinds of projects very differently. The success of a science project is a discovery, which is often a publication, and where is it published, and where you get asked to speak about it, and how many times does it appear in the citation literature, and so forth.

The kind of way most university people are judged, and the way people at Bell Labs and Yorktown and Garcia Princeton, the great Philips labs, the great labs of the world at that time, Siemens labs, that's the way they judge their science people. If this work was at a university, would this guy get promoted because of it?

Not hard to do, but it's a different metric than if you set out to build a faster compiler. Well, did you succeed? And why didn't you? And did it come in on schedule, and did it get picked up? And all of those kind of criteria. There were not very many of the science projects.

House: Oh, I would guess very few.

Birnbaum: One of the things that we learned at IBM-- just by looking that I learned-- but it was really part of the culture there. It wasn't when I first started there. It's a very, very, very rare project that can do both at the same time. You have to be in one world or the other.

Sometimes you can have a science group or mathematics group attached to a practical project, and they can be working on the next generation or they can be proving that what you're doing is mathematically complete or something of that sort.

But I found that most of the existing HP Labs projects that I inherited were both. People were trying to publish university- type papers, and then there was a bunch of people that sat right next to them that, as soon as they published the paper, they would try to implement it.

But implement it didn't mean something that you could pass off to the divisions. It meant built a little-- like a baby prototype, which held most things constant and varied one thing.

So, the result was that I spent the first few months of my life here, at the request of Barney, going to every division review, every annual review, and visiting almost every division lab in the company.

And one of the questions that he asked me to ask them was, how happy are you with HP Labs? What do you wish was different? And so forth. And it was quite astonishing. It was a totally schizophrenic experience.

When you visited almost every instrument division and the medical division and the analytical chemistry division, wild praise. Something like a third of the managers in those triads had spent time originally at the labs, have gone over there with some project and then had gotten promoted, because they were among the most productive people. They just loved the labs. They could point to things that they were counting on, things that they were hoping for, things they have done together, people that they were sharing, etc.

It was fantastic. I never imagined, having only worked in IBM, where it never was like that. Ever. Not even one place I could think of. I never believed that there was a company that would have that kind of a wonderful symbiotic, totally friendly, totally respectful relationship.

And when I went to the labs, and I talked to the labs guys about the division people-- and instruments, they're all part of HP Labs-- I got exactly that story. They would say, oh, fantastic. You know, we gave them this thing, but then when they put to manufacturing, they figured out clever ways to make it cost half as much and be half the size.

I mean, it was just total openness, and so on. Of course, nothing is perfect. There was some grumbling. There were some people that had not such good experiences. But, by and large, it was better than you could possibly imagine. And a very large percentage of the products of the instrument divisions, not the current versions of them but the genesis of them, had come from one of these efforts at the labs or joint effort with the labs.

When I went to the computer divisions, they basically said, "What HP Labs? Never seen anybody from them down here. We've never taken anything from them. They're completely irrelevant. They're a total waste of money. We should shoot it." And, by the way, that's a direct quote from Dana Seccombe who was at the time the head of the VLSI center in Fort Collins.

I heard this everywhere in the divisions. "They work on this crazy stuff. They come down here and they make fun of us. They tell us we're primitive. We can't stand them." Not only was there not good technical relations, there wasn't any particularly good personal relations. In some cases, they liked the people, and there were exceptions.

Then I'd go to the labs guys with these projects that mostly were too small, mostly were too far out, mostly were mixtures of science and applied, and I'd say, "Well, you know, who you working with in division?" "The divisions? They don't even know what an object is. They don't even understand what ergodic theory is. They don't what NP complete means. They don't know this, they don't know that. Encryption? God, the whole division's encrypted."

They had all sorts of nasty, horrible things. It's hopeless. I said, "Well, come on. You're supposed to be an applied researcher." "Ah, we're just going to be so much better than them, we'll stuff it down their throat when we're finished." I'm really not exaggerating.

House: Is that right?

Birnbaum: I had this funny thing. Bill Terry, our head of the instruments in those days, and something of a wag in his annual speech, once characterized to the general management the general strategy of the computer division as ready, fire, aim, and of the instrument divisions as ready, ready, ready.

Because, since they were in a dominant position in almost every business-- another cultural difference. We'll come to that-- since they were dominant in almost every instrument business, either number one or number two, because they'd go out of it if they weren't. It was sort of an early Jack Welch philosophy, almost, that it was more important to have the next generation be really good and really checked out and to uphold the reliability than it was to rush it to market.

So if a project went on for an extra six months or an extra year, but then when it came out, boy it was solid, and it was better, and it was this and it was that, you were already the leader. And you were essentially just improving something that was already the best, so who cared? That was the ready, ready, ready stuff. And the HP Labs guys, perfect partners in that. They're very good at looking deeply and taking their time.

Whereas, in the computer business, as I had learned in IBM, my God, you miss a technology cycle, you're almost out of business. So you're schedule-driven and cost-driven, and the serviceability and warranty costs could completely eat up the products, unless you've built that in properly from the beginning. And here we've got a bunch of guys that are just-- 100 guys are taking on the best minicomputer and the best mainframe manufacturers in the world that had 10 times as many people and that were systems companies, besides.

House: I hadn't thought of it in the terms you've just described it, but when I came to corporate engineering, I found almost exactly the same thing. And my experience in the divisions, we got a lot of help on our scope side, but I'm doing logic analyzers, and nobody at the labs knew what we were building. And all the computer divisions loved us, because we were building tools they could use. Right?

Birnbaum: But then you've got Tom Hornak and all those guys that worked on some of the--

House: Well, but they worked on digital scopes. They didn't do logic analyzers. So, I'd get these, "What's the lab done for your logic team?"

Birnbaum: Then we built with you the Logic Analyzer to simulate the--

House: But we built a nice one for the spectrum--

Birnbaum: Yeah, for the spectrum, where we could actually run the operating system on it. But, anyway so--

House: I've never heard anybody say this.

Birnbaum: Well, I killed all those projects. I killed every single one of them.

House: Oh yeah, well great.

Birnbaum: I mean, they had this project called Bridge, which was a terrible computer. They had this AI database. It was awful. They had two languages. They had a software development environment called Eden, which, believe me, wasn't Eden. And on and on and on.

People were quite good. They were not-- I don't think we got rid of more than just two or three of those people. They were really good people. They had good educations, and so on, but they were working in a vacuum. They were not encouraged to go to conferences. Mostly they didn't read the literature.

They had very, very few joint projects with people in universities, or if they did, they were the worst of the university- kind of projects. The kind that generate a 20 year career and never produce anything in the middle, other than papers. Paper machines.

So, there was a lot of restructuring, and I kind of decided that there were a few guys that could be managers in the new world, but that I had to hire a new core of managers.

And so I spent a lot of my first two years hiring people. I gained a lot of weight. I had breakfast, lunch, and dinner almost every day. If it wasn't with a division guy or a customer, which was rare, I would be recruiting somebody. We hired 250 people in a little under two years. I think they were really good. Typically we looked at 10 resumes to hire one guy.

House: And you personally interviewed most of these 250?

Birnbaum: I interviewed every one of them.

House: Really.

Birnbaum: Up until we got up to maybe 100, 125. I interviewed everyone that was a candidate for management. And the first 30 job offers that came to my office, I rejected.

House: First 30.

Birnbaum: Rejected. And finally, a group-- I remember them quite clearly, I could even tell you the names of people-- they came in a group. They were scared. I tried to put them at ease. I was unsuccessful.

And they said, "We don't understand. We've interviewed these people. We think they're really good. They're getting a Ph.D. from this and this and that. We want to make them an offer. Why are you not approving these?"

And I'd say, "They're not good enough." And they'd say, "They're not good enough?" I said, "Yeah, they've got to be better than you." And that really-- this pall fell over them. "So what do you mean, better than us?"

I said, "You'll never get anywhere if you just hire people that you're better than or that are as good as you. You've got to hire people that you think either are already or have the potential to be better than you. You got to trust them, and then, after you trust them, they will make both of you successful. That's the key to talk to people who've been successful. That's what made most of them successful."

House: Always.

Birnbaum: "And these guys aren't better than you. They're like you. Some of them aren't as good as you. And that's not criticizing you, it's criticizing them. I say to somebody, "What do you think is coming next in computing?" And he says, "I haven't had time to think about that. I'm in graduate school." I don't want that guy in my lab. That guy's not smart enough. Sorry. He's not visionary enough. The best he can be."

The other thing was, I said, "I want two kinds of people here, and I only want two kinds of people, and not one of these people fits into either of my two categories." I probably should have told them all this ahead of time. I thought I had. But I wrote it up. I actually wrote it down.

I said, "I want two kinds of people. I want what I call a super implementer. He can have a Ph.D. or not. I don't care what his background is. But this is a guy, when you give him a circuit to do, or a memory to design, or a switch to conceive, he's not only going to do it for you. He's going to figure out how to make it faster, smaller, cheaper, better. He's going to bring innovation, experience, and creativity to making something." I don't mean manufacturing, I mean prototyping something.

And I said, "These are special, special people. These are really special people. And it's the other guys that I want, I want somebody that you think's got a chance of becoming a visionary, or least having a completely out of the box idea. I don't care if he's got the idea. That's not a criterion for hiring him. I want to see something in his recommendations or in his history that this guy has done something other than just to repeat something that someone else has already done, maybe with a little more accuracy."

Of course, we didn't get-- everybody couldn't do that. But, by and large, we got a lot of people like that. And once we got up to critical mass, they would bring in some other people, and so on. But we made it a very hard gauntlet to get them through.

And for a long time, I didn't trust the guys that were there. Once some of my IT guys-- like Bill Worley and Ira Goldstein, and Frank Carrubba, and Jay Richards, and these kinds of guys who had 10, 20 years of experience and had worked in places like PARC and IBM and so on-- I didn't need to see everybody.

Although I usually would, because almost everybody that we were competing with would have an offer from all those places I just named, and sometimes other ones.

And then we started European recruiting. We said, "You know, this is crazy. Why are we hiring only Americans? It's a big world out there. There's all sorts of Asians and Europeans that would love to come and live in California. And, "yeah, but oh my God, it costs so much money to fly over there." And I'd say, "Come on." Can't spend enough money. This is the most important thing we're doing, is building a core."

So the beginning year was sort of like that. But we got started right away in working on this.

I felt I had maybe three things I wanted to try to accomplish. One was, I wanted to start doing some of the science projects. I felt that you couldn't just live on what you asked universities to do for you or what you read in the literature, because then you're automatically behind.

So I didn't want to work on cosmological relativity or on DNA synthesis or something. I wanted to work on those aspects of computing where there were a lot of unknown issues that would yield to a-- for example, formal methods use in verifying microcode and other aspects of the technology.

But we didn't have anybody that did formal methods. We didn't have anybody that did optimizing compilers. Most of those guys were mathematicians, or many of them were. Topologists, or whatever.

And so we started hiring some guys like that. And, for a while, they were fish out of water, because they were mixed in. But they were always the guy that, while he had theoretical talents or fundamental talents, he really wanted to see something come out of it. So he liked being associated with the group. Well, we measured him differently.

And the second thing I wanted to accomplish was, of course, the thing I'd been hired to do, which was to try to combine the three families into one, which would be a cost leader, the thing which IBM had rejected as not needed, but which had to be really different. I couldn't just give them anything from IBM. They had to reinvent it themselves, under suitable questioning for me as to why they would go at it that way.

And then the third thing I wanted was I really felt that my long, long held dream-- 1978 was when I started really writing about it-- of a computer which would be connected to a network, which would eventually become a national or global infrastructure, and which would be more like an appliance than a computer, and which would bring a combination of computing and measurement and communications to one.

Where computing would be loved by people, would have an interface that was not user friendly, a term that usually produced nausea in me, and that wasn't user centered design, which I hated also. Because what you really were designing was not what you should have been designing-- was not the interaction of the person and the screen or mouse or whatever in front of them.

It was the entire experience that was involved in the problem you were trying to solve, the people you had to interact with, the environment you are doing, and the particular coordinate in space-time that you were at. Because time-driven design was particularly important to me. Particularly in the company that did real time, for goodness sakes. I couldn't imagine that they wouldn't--

House: Was that what you called pervasive computing?

Birnbaum: I called that eventually pervasive computing. I had called it pervasive computing at IBM and gotten joked. Years later, a very smart guy at PARC named Mark--

House: Stefik?

Birnbaum: The guy that did ubiquitous computing.

House: Mark Weiser?

Birnbaum: Yeah. He did ubiquitous computing. It had many of the same things. He and others accused me of stealing it, because we'd never published any of this stuff.

House: Pervasive was a better word than ubiquitous.

Birnbaum: Ubiquitous was the wrong word.

House: Remember the painting that you commissioned?

Birnbaum: Yes.

House: Tell us about the painting.

Birnbaum: Well, let me come back to that. I thought a lot about this ubiquitous pervasive thing, and I tried to get ubiquitous changed. And I did succeed, but strangely enough, only in IBM, but then IBM introduced it.

To me, the notion of pervasive had two principal criteria, to say that something was pervasive. One was that its absence was more noted than its presence. And the second was that was used by ordinary people without special training. Didn't mean they didn't have to learn how to use it, but they didn't have to go for a three year degree or something. And valued. Because if you don't value the technology, you don't build any infrastructure to support it. Ubiquitous meant found everywhere.

So, there's lots of things that you can find that are ubiquitous--

House: That don't fit the criteria.

Birnbaum: --that are in no way pervasive. They have not become part of the everyday life of ordinary people. So, the telephone is surely ubiquitous, and today the mobile telephone is becoming ubiquitous and pervasive. But, there are lots of other things that are very common but they're not everywhere. And you can think of many examples if you put your mind to it.

So, you asked me a question. Would you repeat that?

House: So, the pair of questions-- you wrote a paper toward the domestication of computing.

Birnbaum: Yeah, that was when I first came here.

House: But you also commissioned a painting.

Birnbaum: Yes, actually Frank Carrubba did that. Yes.

House: Did Frank commission it? I thought you commissioned it.

Birnbaum: No, Frank did.

House: OK.

Birnbaum: I gave Frank the good job of popularizing it--

House: Those two caught for me the image of what you were trying to say. OK so the idea of

House: Maybe wrongly, but that's how I viewed it.

Birnbaum: Well, I'm glad to hear you say that. So the domestication of computers was a phrase invented by a very brilliant guy who became like an HP Fellow later, Egon Loebner, that was the way he thought of this problem.

And I thought it was really good, because it related to domestication of animals that went back thousands of years. And essentially it means to tame, or to make them suitable for everyday use by ordinary people. So that's what we were trying to do with computers. Not the Wild West with a software cowboy.

So, but, on the other hand, it didn't tell you what you were going to do, and I thought pervasive was a better term. But domestication had sort of sex appeal.

So, we started a whole set of projects that were independent, which all had the necessity to look through this lens of domestication. Because, if we're successful, with the RISC computer and with the network and the measurements, we're going to want this stuff to be done by grandmas and housewives and people who are retired accountants and so forth.

How are we going to do this? So, we found a lot of different ways to do it. And several of them were very comical. We had lots of talks, we have workshops, we had so forth. But one of the things that Frank decided to do was to commission a painting.

And he had been much struck by a series of ads for a Mexican beer, Dos Equis, that were done by an artist who lived in Berkeley, whose name was Bruce Wolf. And he drew these very realistic Norman Rockwell- style things. And you may remember them-- there would be a guy in front of a firing squad or something like that.

But he also was a very talented sculptor. If you go to the opera house, there's a picture of Kurt Herbert Adler in the lobby-- sculpture-- and that's done by the same guy as well. So an outstanding local artist.

So Frank sought him out, went to see him, gave him a copy of the paper, and said, "We'd like you to read this, and we'd like to commission a painting from you for whatever this evokes in your mind. You choose what it should be." And I didn't know anything about this. It was going to be a surprise for me. And he did that. He interacted.

And then I got a call from this guy, Bruce Wolf, who said, "You know, I happened to have read your paper." He didn't tell me what he was doing. He said, "I'm an artist and I'm thinking about it." He says, "Just wondering, I need some-- what's your wife's favorite flower?" I said, "My wife's favorite flower? What's that got to do with--"

He says, "What's your wife's favorite flower?" And I said, "Well, it's funny. I just bought her some." And she really loves rubrum lilies, which is these sort of stargazer- type pink and white lilies. He said, "OK, thanks a lot." And that's the last I ever heard of it.

And then they delivered this beautiful painting. And what it was supposed to show was a woman in a very bright, sunny, San Francisco, looking over the bay apartment. On the desk is a vase of rubrum lilies. That's why they're there. There's a cat at her feet, and she's reading the newspaper.

But on the corner, just barely visible, is a terminal with icons on the screen. Well, icons only existed in the most advanced prototype in Xerox PARC in those days. And we really wanted to draw a flat screen or something, but it had to be something that people would recognize. So it looked like a computer terminal of the day, but thinner.

And then we had a wine label made, and we had 1,000 bottles of wine, which I was going to give to everybody in the labs as a Christmas present with that picture on it. And we had a poster made for people in the labs and others, and we had Bill and Dave sign it and so forth, towards domesticated computing.

And I thought everybody was really going to love this, because I thought it was a beautiful painting. And I gave it to John Young to put in the boardroom, because I figured, this is good. This'll now become a thing.

House: Sure.

Birnbaum: So, John decided, for whatever reason, that he'd rather have a Japanese abstract along the wall in the boardroom. And it wound up getting put in one of the side rooms, not in the boardroom, where hardly anybody ever saw it. But it was in there, until I retired, when they gave it to me.

And I got about 10 hate emails from people who didn't believe in alcohol that wondered how I could possibly have given wine as a Christmas present.

And then I got one of the posters back from a delegation of five women, in which they had taken this woman and cut her hair out and put little plastic curlers in her hair. They brought me the note and they said, "If you think we are working in HP Labs to provide a haven for some rich aristocrats on Knob Hill to have a computer on the desk, we're not sure we want to work here anymore and we'd like to talk to you about it."

So that was a reinforcement of the lesson I learned that you can't make everybody happy.

House: Isn't that the truth.

Birnbaum: Don't even try. But it was very successful. The other thing that we did was, Ira Goldstein, who headed our software at the time, not wanting to be outdone by Frank, who had produced this commissioned painting, had built for him an eight or 10 foot long radio controlled dirigible.

It was about 10 feet long and maybe three feet high, filled with helium or something. And you controlled it with a radio. And all along the side of it was written, "Domesticated computers are coming," or something like that, of the sort that you would see in a barnstorming blimp.

And he decided to fly this in the high ceilinged building three, where Bill Hewlett and Dave Packard had their offices. And you know the rest of the story, and that it lost control, and it sailed into the executive office area, and it crashed into the tree outside Dave Packard's office just as he was exiting.

And it had a string hanging down from it, and so he grabbed the string, and we had the guy from the HP Labs newspaper there, and he took a picture. And that year at the general management, it was full of pictures in which there was Dave Packard holding over his head the blimp of the domesticated computer.

His secretary, who was a locally famous acerbic character named Margaret Paull, running and screaming, "How dare you? What are you doing? You could have killed Mr. Packard! And whose is this, after all?" And so forth. So the domesticated computer got off to a rocky start. And before long we changed its name to the pervasive computing.

And after a while, it occurred to me, as I was giving these talks, that there was a different way of describing the pervasive computer. And that, using just a catch phrase, I decided to call it mc squared, where the m stood for measurement and the c stood for computing, and the third c stood for communication (MC^2).

Because that's what the company really was. We were the only-- and I said, "We have a terrific advantage, because just using cooperation among our own divisions, we can put together new applications, maybe even new markets, that involve the combination of these three vectors, all of which are undergoing revolution in the world at the same time." The fiber optic thing was exploding.

House: Still could.

Birnbaum: Pardon me?

House: They still could.

Birnbaum: Still could do it. You could do it by partnership. That takes legalities, and so I said it's just beautiful. So I propose this thing. And we started drawing charts, and I presented it at the annual meeting. A lot of people liked it. I presented in different divisions where I'd been invited to talk.

And I found that a lot of people in the product groups especially, as well as in the Labs, said, "This is really great. We could do home health care. We could do a new type of computer chemistry lab on a chip and have it be wireless," and on and on and on.

Most of them were cooperations between our computer division and at least one instrument, or measurement test type division, and our networking skills. We didn't sell networks, but we were one of the largest microwave suppliers in the world. We had many other kinds of things. And all of our computers were networked. That was the nature of what we sold, was distributed computing.

So it seemed like a really good fit. And the company really loved it. And I had a lot of people that started classes. We hired some business school type consultants from Stanford and elsewhere. We did this thing that was called "Create the future," in which all the divisions would come in and they would decide and figure out what kind of things they could do together.

And there were immense surprises as to what people came up with. Because once you turn their heads loose-- in the meantime, the internet had come to be, of course. This was in the '90s now.

House: Yeah, so you've just moved forward--

Birnbaum: I've moved us forward like five, seven years. The company was a successful computer company. We were number three, number four, but we were growing faster than everybody else. And we were really causing both DEC and IBM fits at the time.

So, now I thought, OK, now's the time to really unleash what I really came here for. And it was really popular. I mean immensely popular with all the engineers, all the workers. Marketing guys thought they had died and gone to heaven. They thought it was so fantastic that now they could go in and sell a unique solution.

So one of the early things that we did-- some of the work in the labs, some of the work in computing, you probably know about it-- was the auto diagnostic system with Ford. The thing that was the first system that plugged into the car. Beautiful. Made a giant business out of it. There were several others.

The thing I didn't realize is, I had this very nice diagram, which was a three axis-- measurement, computing, and communication. And at various intersections of the space, depending upon how far along you had to wait till something actually came to be-- lower cost memory, a faster processor, there'd be an application. And the group of people in the labs thinking about it.

I started a marketing dynamics group in the lab that was not composed of MBAs, by and large, although there were one or two to lend some substance to it. But they were mostly senior technical guys who thought this would be more fun than what they were currently doing. And they were really good. They would study a market, they would study how something was currently being done, and they'd look to see whether this mc squared thing (MC^2) could replace it. And so the company--

House: So, in effect you were doing research on markets instead of research on technologies.

Birnbaum: We really did a huge amount--

House: But people who were technologists.

Birnbaum: Rick Corbin was the guy that headed that. Carl Gallivan was the guy before him.

House: Oh yeah.

Birnbaum: I hired him to start that, and he eventually left.

House: Who led it then, after--?

Birnbaum: Rick Corbin, who'd been Vice President of Marketing for DEC. Excuse me, not for DEC, for Apollo. We got him when we got Apollo, and I grabbed him. Super guy.

And we found out all sorts of very surprising things about the way things were done, but the way they could be done if you only had-- and then we would start a research project to see if you could get it. Not all of them were successful, but a lot were.

And then the problem was, we had half a dozen of these that were ready to go. And I had to take them up to the executive committee, which I was now a member of, and which was now run by Lew Platt, to form the organization that would do it. Because it didn't fit in one organization. It fit, by design, across organizations.

And here would be Ned Barnholt, sitting, representing the instrument divisions by that time, and Rick Belluzzo, representing the computer divisions, and the network, both.

And Ned in general was my conspirator. We were always figuring out how to turn his test and measurement business into systems businesses, because he recognized that he could never grow the way he wanted to just selling individual products. He had to sell solutions. To sell a solution, you've got to sell a system. And you need a different set of skills. And you need computers.

So Ned was always super supportive, and Rick Belluzzo was over his head, fighting the PC wars, fighting the RISC wars, because Sun had come along, and MIPS had come along, and IBM now had RISC machines, and so on. And he'd inherited all this.

He was himself an accountant. He knew nothing about technology. He'd run an inkjet business very successfully for a while. And he just got floored by the whole thing. And he just said, "No can do. I can't do it. I don't have any people to spare. I can't put my resources-- I'm dealing with Dell on an hourly basis, and you're asking me to start a remote medical business that won't break even for four years. No, I can't even think about it." So the computer guys opted out.

House: Checked out.

Birnbaum: And so I said to Lew and to others, "Well, if they won't do it, can I go to IBM? Can I go to Sun? I mean, we'll still get the instrument business. I mean, I can just tell you, if you don't do it, there's these new companies coming, like National Instruments and so on, they're going to eat your lunch. Because they're going to do it. And soon all the partners are even going to be gone.

So you don't really have a choice as to whether your instrument company becomes a systems company. The question is, do you do it internally or do it externally? And if you do it externally, who do you want to do it with? So, would you like me to start calling people I know in other computer companies? And now, you're going to be in the situation where you, an instrument group, is aiding and abetting the enemy of the other part of the company."

And they wouldn't do it. They just couldn't do it. They couldn't-- with one or two exceptions-- they just couldn't see the way to do that. So, eventually a meeting came--

House: So the board turned you down?

Birnbaum: Not the board.

House: The executive committee turned you down?

Birnbaum: Yep. There was a strategic-- once a year strategic meeting of the executive committee. This particular one was held in downtown Palo Alto at the Stanford Garden Court Hotel, whatever that thing's called-- the one on Cowper Street. Upper room.

And Lew started the meeting by saying, "I'm going to ask each one of you"-- or roughly 10 people in the room-- "about mc squared (MC^2). Joel has promoted this to where it's almost become synonymous with the company. We've got division managers, people that work for you all, that are out there talking about solutions like this and this and this."

And he said, "We're not really working on any of them. It's just empty, because the computer guys can't be spared, or Rick doesn't think they can be spared. So, I'm here to ask you whether we should just, once and for all, say we're not doing mc squared." So, I was sitting right next to--

House: Did you know this was going to happen?

Birnbaum: No. It started the meeting-- it was a two day meeting. So I was sitting at Lew's left side, and he started the other way. So, sitting next to me was Ned. So Ned and I went last. I remember it really clearly.

And so it went around the room. And everybody would kind of look at me because they all knew I was the one who had proposed it. Pervasive computing, they'd been hearing about it for 10 years.

They said, "Well, it's a really great idea. My people really like it, and I know they really want to work on this, but, oh my gosh, we don't want to work with IBM, and we don't want to work with Apple, and we don't want to work with this. If we can't do it with Rick. We just don't want to do it. So, I guess the answer has to be no."

And it went around to the various people that were there, and then it came up to Rick. And Rick said, "Oh my God, just let me tell you what one of my days is like." And he recounted what I'm sure was a nightmare of 20 hour days and fighting a million fires and trying to hold off IBM and fight with DEC and do this, and do that.

And he said, "Take on chemistry lab on a chip to do gene sequencing? What?" He said, "Who's going to do that? Where am I going to get the slots? I'm losing money on printers, I'm losing money on scanners, I can't do this." Told this tale of woe. "Absolutely not. Cut me out. But, if you guys want to work with someone else that's our competitor, it's fine with me."

And then it came to Ned. And poor Ned was stuck in the middle of all this. He really believed in it, but here were all of his friends and partners and so on. And he said, "Well," he said, "my guys are 100% for mc squared if one of the c's-- or both of the c's, but at least the computer part of the c-- is HP." He said, "Maybe we could go to other people for networks," but he said, "the c is got to be RISC machines in HP, or our PCs, or whatever is appropriate. So, you know, I love it, but I can't do it."

So, then, it was something like eight zip. Then he said to me, "So, well, what do you think?" I said, "You know what I think."

House: Yeah, this isn't hard.

Birnbaum: I said, "Look, I don't care whether you believe in a computer utility, whether you believe that business is going to be online on the internet or not." Time frame for this is about '98, '97. Early days still.

I said, "But, we had a five year head start on pervasive computing. It's not even really a head start anymore. We're maybe a little bit ahead of some companies, and maybe not so far ahead as others, but they're coming up with [INAUDIBLE]. It doesn't matter what you decide."

I said, "You're not doing it for me. I'm going to retire in a few years." I said, "If you want to become a systems company, you have to have appliances, or you have to buy them from a partner. And you have to be part of the computing utility. And the internet is going to have sensors and actuators on it, not just information sources. It doesn't matter whether you want it to, it's going to. You can wait and see and try to catch up later. It'll be hard. You're pretty big now-- $35, $40 billion by that time between the printers and the computers, we've become a very big company. Or, you can keep it going."

I said, "I would recommend to you, at the very least, we keep something going on in HP Labs. Because, if you don't, you'll have nothing. But even if you do, let me tell you that those guys are not stupid. And if they see they've got no customers for what they're doing, they're going to leave. They'll go to some other company that really values what they're doing, as opposed to somebody that gives them a nice salary and good working conditions and says, "Nice work, sonny, but we're not working on that this year."

So, everybody sort of made-believe they were sad. And they said, "OK, I guess that's the end of mc squared then. So Joel, you'll revise your talk, and you won't mention it that way anymore, and you won't do that anymore.

House: That's what Lew said?

Birnbaum: Yeah.

House: The irony of this-- it's exactly the analogy--

Birnbaum: So the reason that people don't know about HP's role in creating the cloud-- we had a cloud five years before anyone ever called it a cloud. We called it a computer utility.

We had a 200 person division under Rajiv Gupta, Caltech, on this, in Cupertino that I had transferred that ran that. We had Karp doing the research-- we had super stuff. I mean, all these things that people can't believe-- gee, start the project on your tablet and you go home and you continue it on your PC and then you use your laptop. I was doing that in the '90s on our product.

But we weren't smart enough to call it the cloud. We called it the computer utility. And the word utility was the worst possible word we could've had, because it reminded everybody of the power company, which is both regulated and monopolistic and backwards and hard to deal with. Or, if you don't like that utility, think of railroads or think of something else.

But we started selling them. And then, unfortunately, mc squared died, so a lot of the charm of that went away. And after that happened, since I couldn't talk about it anymore, I couldn't write about it either. So I never really wrote a paper, other than all the internal ones, on pervasive computing. I certainly never wrote a paper on mc squared, even though I mentioned it in talks all the time.

And so people forgot what HP Labs had done. And the people that got the credit for all these things were not the guys at HP Labs who were really the pioneers in most of them, but other people.

And when Carly Fiorina came, and they had asked me to stay on in the consulting role-- I was already retired as so-called chief scientist, a title they made up-- in which I had nebulous duties, to put it mildly, and I told her this whole story. And she said, "Well, do you think it's too late to do it?" And I said, "No, I don't think it's too late, but I think you're going to have to--"

The thing that was wrong with my drawing, and the thing that was wrong with my thinking, was the same problem that I had when I tried to understand the HP culture after the IBM culture.

I was a technologist, and I thought about technology. And all I thought about was, well of course we can build the computer. It's going to be hard. And then, once we build it, it'll be a fantastic instrument controller and it'll be a great real time computer, because it executes all these instructions in a single cycle, and so on and so on.

I never thought about, oh my God, I'm going to have to get the instrument guys and the computer guys to be friends. And what about all those printer guys? Oh my God. So, I didn't think about how hard it was going to be to create an organizational structure to support the systems business in a company that had never been a systems company.

That we survived that is really kind of a miracle. And that's another separate story we can go into, if you want. But then, having learned that lesson, for me personally, very close to a near- death experience, death in--

House: Now you're back talking about the Spectrum HP story and what--

Birnbaum: Yeah. That was the RISC business, which the business people say and they came out with the RISC machine. Yeah, well, it wasn't so simple.

Having survived that, I made an even worse mistake with the mc squared and pervasive computing. Because the picture that I drew, and the intersections that it had, were just technology. Oh, God, of course we could build the right kind of an instrument with an on-board computer that had the same architecture as the one which was back at the ranch, which was the same as the one that was on the network.

And we could do all sorts of wonderful things. And the guy using it-- user experience design-- he wouldn't even know what was doing the computing. He would be doing speech, and the thing would be recognizing his speech, and it didn't matter that it was a recognition computer of a very special highly parallel sort, with special hard wired stuff that was on the network.

He didn't care. He didn't have to know. The thing in his hand was basically an empty box with a microphone and a speaker and a network connection. And everything else is being done elsewhere.

Who runs it? Where does the profit go? Who services it? How do you advertise it? Whose budget does it come out of? Who takes the responsibility if it fails? Who gets the credit if it succeeds?

Which is worth more-- the computer, which is used in its garden variety, but nevertheless has to be in there, or the front end, which somebody had to invent all sorts of technology for? Who's the hero of that? Because either one without the other is not worth it.

So, there was all these sociology questions. And I started hiring organizational people to help me, and so on. And someone-- I hope he is in a very hot place. He's surely out of purgatory by now-- invented a notion which HP in its inelegant way called rows and columns. Do you remember that?

House: I do.

Birnbaum: And the idea was, this is not IBM. This is not Amdahl. This is not General Electric. This is HP. And HP, everybody likes each other, and everybody trusts each other, and we're all friends. So, we don't have to have a big boss that's in charge of it, and we don't need a big central organization to do all this stuff.

We can just make a matrix. And the rows of the matrix will be the technologies that are required. This technology, printer technology, compile technology, computer architecture, et cetera, et cetera. 10, 15, 20, 25 rows. Those things which had been the separate divisions of IBM.

Well guess what. By this time, HP was a systems company, so we had those kinds of divisions too. What a surprise. As did every other computer company. It didn't matter if you called them divisions. You had an organization that was responsible for that, and it had to integrate with all the others. Well, the rows were the applications. Excuse me, the columns.

House: Columns were applications. Right.

Birnbaum: So here was the compiler, and here was online business processing. Real time data acquisition, real time lab automation, on and on and on and on. There were many medical businesses, analytical-- all the businesses of HP, of which there were so many. 50, 70, 90, who knew.

And there'd be an intersection. So there would be a chairman of the row and column matrix strategy development. They appointed me. That's why I want them to be in purgatory. And I had to preside over a room that had essentially either the division managers or groups managers representing the rows and columns.

So here would be a typical kind of an interaction. We'd be discussing what computer languages should be on our computer families. And the business person would say, "Well, COBOL of course. Half of our customers run COBOL." And the real time person would say, "Well, Rocky Mountain Basic," or PL/I, or this, or C, or whatever was appropriate. And on and on and on.

And so, you would come up with this menu that had almost as many computer languages as there were columns. But you didn't have enough people to do all of those. And you certainly couldn't do an equal job on all of them. So, you say, "OK, we have to decide which are the first four. Well, FORTRAN and COBOL for sure, but now let's talk about the other ones."

And so, you'd have the row guys arguing with the column guy, because, after all, he had a 40 person compiler lab. What was he supposed to do? How could he do this? How was this possible? And didn't have budget to farm it out for other people to do. And so, somehow, all of this was supposed to anneal.

And all of these individual disagreements-- and, of course, I deliberately picked a very easy one, computer language. Talk about networks or talk about operating systems or talk about disk drives or displays. What's the difference between the display between a guy who's doing accounting and a guy who's doing the logic analysis? Big difference. And you couldn't support every display in the world. So what would you do?

So the idea was to do it out of rows and columns. And I did that for something close to a year. There was no decision making process. There was no way to say, "We will do these two or three," because I did not have that authority as chairman. My idea was to influence and coordinate, but not to decide. The decision had to come from within the group, but people wouldn't agree.

House: So it's four peers sitting there, essentially.

Birnbaum: They just wouldn't agree. They were all peers.

And so, individual arguments would seep out of the committee and get before the executive committee. Those experts on the intersection of computer architecture and application. And they would look at the body language, and they'd look at the arguments, and they'd look at the size of the markets, and they'd look at who was where, and who needed a favor lately, and what was this guy's history.

And it was just comical. But it wasn't comical. Took a lot of people's time. Plus, everyone to fly in for the meeting, because we didn't have such good virtual technology, which isn't so good for negotiation anyway, because a lot of it doesn't happen at the table. So we had once a month meetings, and technology was moving at a speed that didn't allow for once a month meetings. You needed meetings once a week or once a day.

And so I made the recommendation that this was a failure, and that we could not do this. And I prevailed upon, I guess it was still John at that time, that we needed an organization. It didn't have to be a permanent organization, but we needed an organization that had all those parts in it, with a leader in it. And that the leader needed to be somebody who understood both the industry and the technology, the rows and columns.

And we put them all together, and we called it the Information Technology Group, and it would be 1,000, 1,500 people. And, once we had gotten over all the hurdles, then you could make the decision whether you wanted a permanent technology supplier to the business units, or whether you want to break it up and put them back where they were when they started.

And they agreed. And they formed ITG, which had the advantage of, boy, decisions were being made. And then they made a catastrophic mistake, or close to catastrophic mistake. You want me to stop? Am I--

House: No, you're doing fine.

Birnbaum: Instead of appointing someone from the computer business who really understood it, or at least was learning to understand it, or going outside, which was my recommendation and getting someone who has lots of experience but who was really neutral-- someone from IBM or someone from DEC, or something like that. Not the highest level guy, but-- They took somebody who was smart, universally well-liked, immensely hard worker, and 100% unqualified. That was George Bodway.

George was a giant in semiconductors. He'd won many awards. He'd made HP a leader in many different technologies. He'd run a marvelous study across the technology. He was a member of the National Technology Roadmap Committee. He could have been a professor at any university in the world that he wanted to. People loved him. And he deserved their love because he was both compassionate and intelligent.

And he knew nothing whatsoever about computer science, or the computer industry. He'd only been a chip supplier and a fab constructor and runner. He was wonderful with that. But that was one row. And they made him the head of it. And they said, "OK, you make all the decisions, George."

And George worked so hard. He hired a lot of good people. All of them had very strong personalities. And they would argue and fight with each other, having the fights that used to take place in the Strategy Matrix Committee, they’re now happened within one place. And they could never agree. None would ever defer to the other one.

And so they would go to George and say, "George, decide." And mostly George wouldn't decide. He would let them both exist. So, suddenly there were nine or 10 different computers under simultaneous development. There were three versions of this language and two versions of that language, and the software that didn't compete.

And it was slipping like crazy. Spectrum was now a year late. Then it was a year and a half late. And then John Young asked me to do a detailed review for him. I was down there all the time, anyway. I saw what was happening. And I did.

And I said to him, "It's gonna fail, John. For sure. You have the wrong leader. You shouldn't punish George. He worked harder than any human could ever have done. He did everything that was in his power. He went to school, he learned, he's a terrific guy. But you have him in the wrong job. Just because he could be successful-- he could do that job, if he had cooperative people underneath him who were willing to work with each other, but that's not what he has. And with all the prima donnas he has-- Denny George, Dana Seccombe et cetera, et cetera, Peter Rosenblatt, Bill Worley, on and on and on. No. So, you've got to replace George."

He said, "Any candidates?" And I said, "I could think of people that I used to know in IBM that could do it. Whether they'd be interested in coming-- it'd take them a long time to learn the company. I'd take one of those guys that's down there and promote him, and just hope he can do it. If you feel like, I'll look over your shoulder, whatever you want." He said, "OK, let me think about it over the weekend."

And next week, he called me into his office, John Young did, and he said, "OK, we decided to follow your suggestion. We're going to remove George Bodway. He says, "We're going to have a new leader that goes down there. Full time, 100%. And it's you."

House: And it's you.

Birnbaum: It's you. I said, "It's me?" I said, "Why me? I never built more than one of anything in my life. I don't know anything about manufacturing. I don't really even know that much about modern software development. I never ran a profit and loss sheet. How am I qualified to run this massive division?"

He said, "I don't care whether you're qualified or not, Joel. The writing is on the wall. If Spectrum fails, you're going to get the blame, because they're all pointing their finger and saying the architecture's no good. Because they want to find a scapegoat. You're the scapegoat, so you're not going to have your job if this fails. And you have a chance to either go down there and try to fix it, in which case you'll be a hero. Or you can go down and fail, in which case you lose your job, but you would lose it anyway."

House: Put up or shut up, huh?

Birnbaum: So I said, "OK," I said, "When do we start?" He said, "What are you doing tomorrow?" I said, "Tomorrow?" He said, "Yeah, tomorrow." He said, "You and I are going to go down there. We're going to announce it to the people. And we'll set up an office. You start next Monday."

It was like a Tuesday or something like that. I knew I didn't have any choice. I came home, I told Eileen, "Have a bag ready." New York's looking pretty good right now." Because IBM had offered me my job back.

And I went down there and that's when I got my really, really bad shock. I had thought that what he had said to me was that we were going to be co- leaders, George and I, and then George would make the decisions, that I would be there, really making the decisions and feeding George. I think that could've worked.

George and I were friends. We liked each other. And I know he would've accepted it, and I'm sure I could've made that work. And I get down there, and I found out that they were firing George, and I was replacing George.

House: Fire him?

Birnbaum: Out of his job. George had no job. They just walked in his office and told him, "You've got no job, and here's who's replacing you-- Joel." He looked at me like I was Judas, whereas I hadn't had anything to do with it, and he started crying. And it was very, very sad.

And, for about a week, nobody would speak to me. All the people in the office, and all of that, they just wouldn't have anything to do with me, because it seemed clear to everyone that I had done it. And I hadn't.

And I immediately got a group of people that knew their stuff, that had been through this. Principle among them Jay Richards. Plus many, many others who'd done IBM or other development for years and years.

We set up a war room, built it. There were 28 projects we killed. 28 managers-- I fired 25 of the managers. Restructured the teams. Took Dana out of his job, because he was a divisive-- brilliant guy-- divisive force. Made a deal with Dick Hackborn to make him in charge of the inkjet supplies. As you know, we had an immense success doing that.

And made many, many changes. Completely changed what was going on in the labs to be much more applied-- actually, development. Just bringing in the shock troops.

And George they found a job for in corporate. And I had a lot of discussions with them about what actually happened. I don't know if he ever forgave me. He behaved in a friendly way, but that really wasn't my fault.

So, that's when I inherited. And it was a workable structure. I was completely the wrong person for it. I had to learn almost everything that we were doing. I didn't know how to do any of those things.

In something like 2 and 1/2 years, now three years late, we shipped Spectrum, and it was an immediate success. And HP became eventually the number one computer company, even outpacing IBM, a number one technology company. And we almost ruined IBM in the process, and we did eliminate DEC in the process.

And I don't even know if ITG still exists. I think it doesn't. I think they dismantled it. When it was over with, I went back to the labs in the corporate role.

And before that, I did information architecture, which was getting back to the pervasive stuff. That's where we did the architecture for what became the infrastructure. Did a number of things that later became standards, and so forth.

But I think the company never really quite believed in my smoke and mirrors story. They sure believe now.

House: Well, it not only saved the company, it made the company.

Birnbaum: It's the world. It changed the world. I didn't think it would happen so fast. I didn't obviously foresee all the different things that happened. But I was so certain that when people could pay by usage and could substitute a subscription for capital investment, that had to be the same as electricity and gas and water, and so forth. All of which you can get yourself, but it's too much trouble.

And I had run into early versions of it. When we had the utility, one of the things we stuck on our utility was a big Amdahl machine and a high-performance floating-point processor. There were no groups in HP that could afford to have one of those of their own, but collectively, you just signed on and you used it. It was clearly the way to do things. You got charged so much per hour.

And sadly, HP didn't lead any of these things it might have led in. And, in the end, it was done in by its old culture of not being a computer systems company.

And by having a form of benevolent despot as a leader, who was unwilling to make the tough decision of telling one of the equals, no, you are subservient in this case. So you do this and this. They didn't have any trouble doing that as long as it was within instruments or within the computers itself, but when it came to be across them, it was really, really hard.

And, with the death of mc squared, death of pervasive computing in the company and the company not doing as wonderfully as it had been, we had people on the board who said, "Why don't you just spin off the printers? They would have a huge stock value just by themselves. Why don't you just spin off the instruments? Why are you burdening everybody with a tremendous albatross of the PC business?" And the now struggling server business, because they hadn't gone into what they should have done. And they said, "Yeah, why not?" And so they did.

So I opposed the break up of the company very, very strongly, and very volubly. I had been mostly quiet in all those years of the executive committee, except when I was asked to give a talk. I was not quiet now. In fact, I was so unquiet that Lew removed me from the executive committee to avoid disruption.

So I spent the last six years of my career not on the executive-- six months of my career-- not on the executive committee. Supposedly as his advisor, but my advice was never asked for.

So, when I think back on IBM versus HP, IBM really, really believed deeply in research, but it didn't use it very often. Whenever it did use it, it was a big deal, like the one device memory cell, or DDS encryption, or something. There were only a few of those, but they were really major practical, etc.

HP super believed in the research culture, but didn't really support it. And when push came to shove, the first thing that went were all those nice projects that were the third of the effort that was aimed at doing new things, not improving things that were already there. Whether that meant deep research or just a new application of something you already knew, they went away.

Like, for example, all the wonderful medical programs, which resulted in our selling our wonderful medical business to Philips, who, ironically, has done a lot of the things that we did at the labs, like the defibrillator, and things of that sort.

So HP is struggling because it's still trying to learn, I think, how to be both a product supplier and a systems supplier. If you visit Agilent today, they took the bit between their teeth, both in Ned Barnholt and in the successive leaders. They brought in the systems expertise, they formed the partnership, they're in the systems business. It's beautiful. They're rewriting the whole book on how life sciences is done and how you really integrate computers and networks and instruments. And it's quite beautiful.

HP could have done the same thing 10 times over. It still could do it, but I never read about it at all. In fact, under the reign of Apotheker, thankfully short reign, he decided they would go out of mobile devices. Father, forgive them. They know not what they do. But my goodness. What could they have been thinking?

House: So you did serve as an advisor briefly to Carly?

Birnbaum: For five years. For five years.

House: But nothing came of that, either?

Birnbaum: Yeah, something came of it. I wrote a big plan for her to how to sell the PC business to Lenovo and the Chinese. If not them, Acer. Contract them to build private label PCs that would say HP on it when we want it to, but not to have to do the development or manufacturing or marketing expense.

And to use the money to go into what I called at the time genome computing. And because Compaq already was the supplier for that principal one, we sort of inherited that market. And to go into storage systems, which I felt were going to explode and become currency of the new infrastructure. So, sell the PCs, and go into storage and go into--

House: Would you sell printers at the same time?

Birnbaum: No, I didn't think so. I thought printers-- I had lot of thoughts about what printers should become, some of which they still haven't done. But she loved that plan. And then I picked up the Wall Street Journal and then we had bought Compaq. And I just couldn't understand why in the world we would have done that. I jokingly said to her that she dropped a bet that I had meant. I had said sell and she interpreted it as buy. She didn't like that.

And she pointed out how the two customer sets were orthogonal, how we got a strategic foothold in the very businesses we had recommended, the storage and the other, because together HP and Compaq was as big as mc squared, and so on, the storage company, and IBM. IBM

But they squandered it. And they didn't execute. If they had executed the integration plan which Carly had written but which she didn't supervise, and which Webb McKinney actually made the mechanics work right but not the intent. I think it would have worked. I think it would have worked. Even without the mc squared. But they didn't do it.

So, I was there for a while, and then I was about to resign because I was frustrated, especially after they did that. But I kind of felt like maybe I could help them be successful from the inside, and if I retired, then there was no hope.

This earned me the enmity of the ones who had retired. They didn't talk to me for quite a while because they thought I had sided with the enemy, but, in fact, I was trying to change it from within, but I had no success.

And then after 9/11 occurred in New York, a congressional committee asked the National Academy of Engineering to develop a research program in computer sciences that would focus on terrorism and cryptography and so on. I was asked to chair that. Carly wanted me to still be an HP employee while I do that, because she wanted HP's name on it, and she wanted whatever insights came. So I did that for three years.

And Carly left and Mark Hurd came and I was history. So, yeah, I did stay on until 2005, I think it was. Although I actually retired in '99.

House: OK. Well, this is a fascinating story that you've just woven for us. It's not altogether a winning story, in some ways, but the learning that you went through-- I'm struck by the recurrent tension between technology on the one side and the political realities on the other side.

Got in your way at IBM. You've got this great research lab that they said, thanks sonny. You've got this fabulous proof positive that HP Labs and they've got essentially-- and I knew George for 30 years prior to you coming, almost-- and I agree with all of your assessment. All of the positive things and clearly knew nothing about computing. But you pulled that one out, even though you might not be the guy. John had you right in the crosshairs.

Birnbaum: Lot of good guys worked really hard.

House: And then with mc squared, the same basic issue arises again. And this time they punted. They punted as bad as IBM punted. Right?

Birnbaum: Right.

House: And let's just see how the world unfolds, and the world unfolds as you saw.

Birnbaum: You remember that IBM was really on the ropes, and a lot of the analysts were writing them off, almost. Stock had plummeted.

And Lou Gerstner was brought in. And he was from Nabisco. And he was a fantastically smart businessman. A tough guy. I came to know him because we were on some committees together later. And he looked at what had been proposed, which was, what are the things that IBM does well, and let's break it all up and make them into separate companies, and so forth.

And he looked at it and he said, "Oh my God, what IBM does well, better than any company maybe that's ever existed, is take 25 separate divisions with 25 separate things and do a field merge in one day and ship them to a customer who thinks that it's a box that was made in one building. And it runs and the guy in a suit with an attache case fixes it in an hour if it breaks." He said, "We're not giving that up."

House: Yeah, exactly.

Birnbaum: "We're keep that. We're not breaking up the research division, or whatever." And boy, he laid it on-- and the guys that followed him did it even better, and they were more knowledgeable about the business than he was.

And so, if you go to IBM research today, you'll find it's bigger than ever, more important than ever, more value than it used to be, and a lot more of their stuff is making it out apparently, which is good. Certainly it's much more visible with the Watson and Big Blue and all that stuff.

Whereas, research labs at HP, unless it's changed recently-- I haven't been there in a couple years-- they weren't doing research at all anymore. They were working on projects with two and three year time horizons in partnership, by mandate, with the product divisions. That's, at the best, advanced development.

House: So I just heard a talk by Stan Williams. And he described the four times in the last four years he's been ordered to quit doing what he's working on. I mean, it's just shocking. Stan Williams is a guy that's been three times invited to Stockholm to speak to the Nobel Prize committee.

Birnbaum: That's probably the only reason they still exist. That's the best project that I knew in HP Labs when I was there. The quantum structures stuff is fantastic. That was how we started basic research. That was our first basic research lab.

House: So, I could ask a hundred other questions, but at one level, I think we've done a nice job. What would you characterize as the proudest thing you've accomplished in this 40 years? That's almost an unfair question.

Birnbaum: Well, the question, it probably needs to be reworded. Because I don't think I accomplished much of anything. The best that I ever did was to create an environment where people were not afraid to take risk and were failing didn't mean not achieving your objective. It meant not achieving and not knowing why you didn't.

House: I'll accept that. That's great. That's a wonderful answer.

Birnbaum: But, so, I think I encouraged a lot of people to do more than they thought they could. And a lot of them succeeded in a lot of improbable ways. And I fell very, very good about that.

In terms of things that probably wouldn't have happened without me, I have the strong feeling that RISC technology would not have happened the way it did. I think IBM had dropped it and only resuscitated it because of HP and the impact it was having. And then followed by Sun, which did a different version-- Patterson version-- and MIPS-- the Hennessy version. And a couple of smaller companies also picked it up. But the cost/ performance advantages were so great.

I do not claim to be the inventor of that. That surely was John Cocke and his associates. I was, I'd say, the editor, and, more important, the producer. And I think without my editing, it wouldn't have come out as well, and without my being the producer, I don't think it would've happened at all. It certainly wouldn't have happened at HP. And I doubt it would've happened at IBM if it hadn't happened at HP. And if you look at computers today, even the PCs use the RISC philosophy on almost everything.

And so, one of things we've all gotten spoiled with is that we can have a watch or a telephone which has the computing power that some universities used to have. And we kind of say, Oh yeah, well, God, they learned how to make those chips small.

Well, it's not just the chips. It's the software that they run and the architectural style that runs that. And things like the ARM chip and all the things-- the Acorn and all of those other RISC machines that came later and that are so effective in these mobile devices and in the instruments and so forth-- I don't think they would have gotten developed without the work on optimizing compilers, which was way, way, way beyond what those people would've been capable of, or would've been likely to work on.

Because, without that strong theoretical underpinning, which came from HP and IBM and, to lesser degrees, from some universities, built up a wonderful technology.

And now the question is, what do we use technology for the world? How can it help people? You read debates about are robots good or bad for the world? And all that sort of stuff. And so I'm very proud that I was part of making that happen.

Lots of other people had so many wonderful ideas. The stuff at PARC of course was fundamental in all this, and fundamental to Apple, and fundamental to our notion of a machine should look like and how much we should be expected to know. If you had asked me this question 10 years ago, I might have given you a very different answer. Or 15 years ago.

But it was probably the biggest mistake a judgment that I think I ever made. I still don't understand why it was wrong. Maybe that's interesting to just talk about for a moment. As soon as I started thinking about appliances, which, as I said, was in the '70s,

I really thought of them like an appliance. Named by what it does. Washing machine, not electric motor which brings in water, mixes it with suds, and vibrates it. No. Washing machine. Dishwasher. Electric toothbrush. Razor. Et cetera.

What does it do? And then, what's the least you'd have to do to use it? And then how can you optimize it so that does that one thing as well as it can do it at the lowest possible cost, both material cost, and cost of learning, and cost of service?

And the appliance industry has taken that to an unbelievable place. And things that you didn't think of as appliances, like printers. Remember how complicated and how many thousands of printers and appliances. Buy it. Boom. Plug it in. Does its thing. What's it do? Oh, it prints, it scans, copies. We don't think about how it works.

So, as soon as I started thinking about appliances, I realized that there were two ways you could build appliances. And I wrote a paper on them. It was an internal paper at HP. I don't think it ever got published. The name of the paper was, people thought, funny. It was called, "Swiss Army Knife or Batman's belt?"

House: I didn't see that paper.

Birnbaum: I think it was supposed to go in the HP magazine, and I think it didn't go in. Anyway, I don't remember why.

House: Mid '90's?

Birnbaum: Yeah. Yeah. But I'd been thinking about it for 20 years by that time. And it was unclear. So, the Swiss army knife is like the decathlon athlete. The guy that wins the decathlon in the Olympics is called the world's best athlete. In fact, he probably is.

But, in fact, in every one of the 10 things that he does, there's probably 15 people that can do it better than him. There's better high jumpers, there's better sprinters, there's much better milers, and so. But the thing that's remarkable about him is that he's good in all of them. And he's just unbelievable how they can do that. But that's the decathlete.

As opposed to the guy, Edwin Moses, who only runs the hurdles. One event. But he's the best that's ever been, and he spends every minute of every day just working on how he can shave off a hundredth of a second going over each hurdle.

So one way to build appliances is to be the decathlon guy, which I call the Swiss army knife, because I wanted it to be a thing. Swiss army knives-- I mean, I have one at home with about 40 different functions.

It can do everything. It can start fires with a magnifying glass and it can be a tweezer and can be a knife and it can be a corkscrew and it could be a bottle opener, and on and on and on.

Again, it doesn't do any one of those things as good as a tool specifically made to do it, but it's one thing and you can stick it in your pocket or your knapsack, if you have a big one. You know you can do an awful lot and you can probably get by with just about anything.

Then there's Batman's belt. OK, so if you know Batman, at least in the comics, he had a belt. And he was this great inventor, Bruce Wayne, down in his basement laboratory. And he would invent the thing. If he wanted to fly, he'd have a little thing, and like Spiderman he'd shoot a thing on. Or he'd have the Batman belt that launched a ladder.

Every one of those things was an appliance. It did only one thing, and it could do it really well. And it could be simple. And it would be obvious how to use it. Basically turn it on. Because it only could do the one thing. Once you learned what that thing was, that's all you had to learn.

So the question was, here's going to be information appliances, and which way are they're going to go? We're talking about computers. It's general purpose. We could make one thing that would do everything, or almost everything, or we could do something that did one thing.

So, as I think you know, in Bristol, England, we started an appliance laboratory. We started it in 1985. Bristol was started maybe '84. It was the first lab. Bill Sharp was the guy who ran it. And I said, "Go after Batman's belt primarily, but keep your eye open for the Swiss army knife. But Batman's belt is what we're going to work on."

Well we came up with the damnedest things, many of which were actually built, but hardly any of which ever made it out. I'll give you an example of a Batman's belt appliance that we actually could still make and which I think would still be useful.

It's the size of a fountain pen. It looks like a fountain pen. It's got a little computer and a couple special sensors inside. You take it with you to the supermarket, and you hold it over melons and other fruit, and it tells you which ones are ripe by doing a chemical analysis of the ketones that are given off. It does what the guy does when he smells it, but it does it better than him by a lot.

We felt we could make one of those for about $15. The Shopper's Friend. So every time you go shopping, or any time you want to know whether the fruit is ripe, you do that. Well, there's another one that detects salmonella and botulism, and there's another one that does lead paint in your kid's nursery. And there's another one that does this, and another one that does that. $20, $30, $40.

A prototype for thinking that way never had the slightest success in getting HP to do this, was a very wonderful appliance that you own. I guarantee you own it. Everyone I know owns one. You've used it 100 times. If you know how it works, you'll be the first person I ever met that knows how it works.

This is the stud sensor, which finds where to hang a picture in the wall. You take the thing, you lay it against the wall, and you run it along, and it comes to a peak, and it shows you. And you say, fantastic. And you put the thing-- you don't have to dig up your wall.

Costs $15, $20. The guy had 17 years of the patent, and he sold something like 500 million of them. Some giant number all over the world. They had different names. He licensed them to other companies. And now they're many competitors. You can out to any hardware store and buy one. And they have some improvements, but basically, it's the same thing.

So the question is, how does it work? Every says, oh, it's a magnet, it looks for the nails. Sorry. Doesn't have to have a nail. How do you think it works? Well, I couldn't figure it out. It drove me crazy.

So I wrote to the company. And I said, "Would you tell me how this thing works? Because I can't find anyone that knows." I took it into the labs. I showed it to all those smart guys. I don't know. Well, it measures the dielectric constant. That's what you want-- bulk. It's looking for an increase in density.

It's the world's worst dielectric meter. It's good to two bits. It costs about a nickel to make it. And the irony was that HP was the world's leader in making machines that measured the dielectric constant. They cost $50,000. They measured it to nine decimal places. They sold a couple of them a year. Peanuts.

So I started this campaign. I don't remember what I called it, but I guess I would call it for the purpose of this interview, the crappy instrument corporation. In other words, what are things that people really want to do that require extremely low precision, like find the beam in the wall?

We don't have to measure the lead in the paint, we just need to know whether there is any. Because if there is, I don't want my kids sleeping in that room. Is there radon in the house? You can go on and on. Can I measure my water quality myself? Can I measure the air quality?

So we had this entire higher range of the things to go on Batman's belt, and my God, were they sexy. They were small, they were cheap, they have a big light that would go on and say, poison. Or something like, don't eat this salmon, it's poison.

Versus what we invented. It was about two years before the so-called Newton. And it was a thing. It was small. The electronics was in a knapsack that you wore on your back. And it did all these different kinds of things. It kept your appointments. It did a few of the things that smartphones do today. It was not a phone.

House: More like a tablet.

Birnbaum: Yeah. And we gave it two companies, an insurance company and one of the English HP divisions. And we let them have it for six months. And as clumsy and as crude and as horrible as it was to carry around with us, they got so addicted to it, they didn't want to give it up. They wanted to keep this crappy thing.

And its interface was handwriting recognition, like the [INAUDIBLE]. And we had started a group in handwriting recognition. We had formed partnerships with a couple of the universities that were leaders. We didn't tell them why we were doing it, we just wanted to be able to do.

And it was so primitive and so awful that we actually advised the company not to make a product of it, in spite of the very positive user feedback, because they're not ready for it. We either need speech recognition or a better form of hand recognition. We can't have something this small with little tiny buttons that people are going to hit. Well, that turned out to be dead wrong. People are quite happy with the little--but we didn't think of that. So, we ended that.

So, out of that I should have learned, from the fact that these people like that, but they didn't like it in the wild, crazy ways of people like these little appliances, that Batman's belt was going to win. And I said, "It sounds crazy, but I think people are going to sell like a shelf in your house, like a little tool things, and it's going to have all these little things. And when you go to the supermarket you take this one. If you go camping, you take this one. If you go do that, you do that. If you're working on the lawn, maybe you'll take this one, and so one.

Because I can't imagine that anyone is going to put in the time to run hundreds of applications on a digital Swiss army knife. Because, no matter how you do it, the interface is going to have to be complicated and you're going to have to learn an awful lot of special things." And you're back to the decathlon, because it ain't going to be as good at any one of them as anything else. Sorry, flat wrong.

House: Perfect story.

Birnbaum: The Swiss Army Knife wins. It's called the iPhone. I was a user of the Palm Pilot and before that, the Sharp Wizard, and before that, I always had one of these. I hated them so much. I'm a nerd, so I learned all the interfaces and I did all that stuff.

And I would try to teach it to my wife or somebody that wasn't technologist, and they would just look at me like I was absolutely crazy. So every time I got to thinking that people wouldn't prefer the individual appliances, I would try to show them how that worked.

The brilliance of Apple and the wonderful experiences they had with the Mac and everything else, let them use the touch screen and let them have the app and let them do something that I would've thought they could've never pulled off. We could never have pulled it off in HP.

The contribution of Steve Jobs-- many of course-- but certainly wasn't in the invention of the devices. As far as I know, he didn't invent any of them. He probably edited them and so forth, but he wasn't a technologist per se.

And the real contribution that I think he did was to have the strength of will to see these things through for the five or 10 years that it took of political infighting to get digital rights management to let the movie industry do this to do movies, to do this, all that kind of stuff which nobody in HP had any stomach for. They didn't want to spend their time doing that.

And then the other thing was an argument that I personally had with him a couple of times. I believed that no one company could produce all the good solutions and all the good devices, and you needed an open system.

So, I was very involved in the creation of the Open Systems Foundation. As you know, its first president was Ira Goldstein, who I nominated instead of myself, which is what they really wanted to do. And I believe that, while that was hardly perfect, and every time you set up a foundation in an open structure, at least everyone in the world would be able to create fundamental grassroots ideas, and they would all run on any platform that was willing to subscribe to it.

And my hope was that some of the applications would be so attractive that a common platform would arise made by many people.

Well, Steve Jobs said no way. People aren't like that way. They'll never agree. The thing will get to be a big bowl of spaghetti. It's horrible. I will control everything. It will not be open. In fact, it will be the most closed system you've ever seen. Everything will be guaranteed by my company to run with everything else, because otherwise you're not allowed to put it on our hardware.

And I said, you must be crazy. You must be absolutely crazy. Who in the world would ever accept that? And don't you think there are smart people in the world outside your company? Oh, we'll take care of them. We'll figure out a way to get them involved. Which of course he did. That's the app store. I didn't think of that.

And so he stuck to that, stuck to that. He was the bad boy of open systems for years and years. Didn't believe in Linux, didn't believe in Redhat, didn't believe in anything, if he ever would say anything.

But at the same time, he used the world's best Unix, the Carnegie Mellon kernel, to become the basis of Mac OS. So, while he was decrying the open systems, in fact he was using it to establish the world's greatest closed system in which things really work.

They're rock solid, and you have a guarantee, or darn close to a guarantee, that everything is going to plug into everything else. You can do Airplay, you can do this, you can do that. Hats off, Steve, he was really right.

Or, putting it a different way, the stumbling, bumbling, squabbling factions that I ran into at HP were no different than the ones that were between companies or between industries or between groups in the same company.

And so the open system flounders along. Now with Google and all of that I guess they're starting to be something similar to what Apple did. But that isn't an open system, either. You're just conforming to somebody else's sets of rules on somebody else's platform.

House: So to go back to Steve for a moment, he came back in '97 to Apple. It was 2004 before he actually turned the corner. Any board like an HP board would've tossed him long since. You talk about somebody able to be decisive, never mind what the issue is, he can stay the course.

Birnbaum: He was the most-- I think he's one of the great men of history. And I know I often see him compared to Edison, and I think it's a quite reasonable comparison in many ways. Because Edison himself wasn't the guy that came up with most of the ideas. In fact, when you look at some of his most famous inventions, the light bulb and many of these others, they existed for 20, 30 years before.

No one ever figured out how to make them practical and make them cheap and make them long-lasting, how to build an infrastructure. He was the great designer of integrated systems. He figured out how to make it all work.

Sadly, he made a tragic mistake in choosing direct current but you can't win everything. But well, that means he ended his life in poverty in hiding from debtors.

House: Oh did he really?

Birnbaum: Oh yeah.

House: I didn't realize that.

Birnbaum: He had a secret compartment under his desk in Menlo Park in Naples, Florida, where he would hide when the banks came. And he would go under the floor.

House: Is that right?

Birnbaum: Yeah, it's a national park now. He ended sadly. Because the guy that was really his opponent, who was considered the loser, was Tesla.

House: Thank you so much. This has just been a fantastic interview. We will get you transcripts, and we'd like to have you read them and edit for correctness.

Birnbaum: We're done?

House: We're done. Is that OK with you?

Birnbaum: Sure.

House: I'd just like to thank you on behalf of the Museum a lot.

Birnbaum: I don't know if this is of any interest to people.

House: Personally, I'd like to thank you just for the privilege of letting me interact with you.

Birnbaum: Oh please.

House: Just great.

Birnbaum: You're the greatest. I appreciate it.

END OF INTERVIEW