Oral History of Kit Colbert

Interviewed by:
Dag Spicer

Recorded February 12, 2024

CHM Reference number: 2024.0018

Spicer: So here we are in Mountain View at the Computer History Museum with Kit Colbert, an amazing contributor at VMware. The date today is February 12th, 2024. Kit, welcome.

Colbert: Oh, thank you. Thank you for having me.

Spicer: Absolutely. You've done a lot of really interesting things that we want to dig into. Before we do that, though, I'd like to just ask a bit about your early years, your upbringing, your family, your family life, and what that was like?

Colbert: Sure. Let's see. I was born outside of Boston in a little town called Somerville, which they called Slummerville at the time. <laughs> It wasn't a nice area, but now it's actually quite fancy. We were only there for a couple of years. I mostly grew up in Oklahoma. We lived in a couple of small towns, or some large, some small towns in Oklahoma. But I remember getting into computers fairly early on in life. I remember my dad got us-- well, obviously, we had video games early on, Nintendos and all that. But then we got a personal computer, when was it? It would've been when I was probably about nine or ten, I think?

Spicer: What year would that have been?

Colbert: That would've been '89 or '90.

Spicer: Okay, thank you.

Colbert: Yeah. When did the [Intel x]386 come out? Or 486? <laughs> Do you know off the top of your head?

Spicer: You got me there. <laughter> Sorry.

Colbert: It's okay. I'm trying to think, because I remember-- well, basically, what happened is I started playing with these computers, the computer, and I started breaking it, obviously. So--

Spicer: You mean breaking the hardware? Or--

Colbert: Well, software and--

Spicer: -the software?

Colbert: Yeah, digging a little bit in the hardware. But my dad brought me over to the place that built it. So at the time, it was very common to have these really small, independent shops building computers for people. Essentially, I started spending time with them, started working with-- actually, I worked for them over a number of summers, piecing together computers, sort of learning how the different components of a computer worked. So both some hardware stuff as well as the software side of it. So that's what really

got me into it, eventually building a computer for myself and getting me more into all areas of that, like the programming aspect, the actual putting together all the pieces, DIP switches, and all these fun things from way back in the day.

Spicer: So these were all PCs, right?

Colbert: Yes.

Spicer: Yeah, PC-based?

Colbert: Yeah. I was not a Mac guy until probably like 10 years ago <laughs>.

Spicer: Right. Yeah, well, I know for some of us, the Mac was a bit pricey, and that kind of kept us out of the Apple universe, unfortunately.

Colbert: Well, I think I liked the-- at the time, the open nature of the PC, the fact that you could have all these components from very different manufacturers, but they're all compatible in certain ways. So the fact that you could actually very much fine-tune and customize the PC for what you were trying to do, which I think's something I really, really enjoyed about it.

Spicer: Yes, that's true. Although, from the point of view of people who are less computer savvy, I remember those DIP switches… setting the interrupts <laughter> on the card and stuff was like an endless headache--

Colbert: It was.

Spicer: -for a lot of people.

Colbert: I remember reading some of the early PC magazines and these sort of things, and I naturally got into it and got into the DIP switches and all the different jumper cables and all this sort of thing. At some point, I was reading a magazine that said this one card-- maybe it was a 3D accelerator. I forget what it was. But anyway, they said in order to… the technical level of expertise required to actually install it and configure it was "guru" level. <laughs> I remember I was like "Well, that seems kind of intimidating," but I went and did it. So that kind of gave me some confidence, "Oh, maybe I could be good at this whole computer technology thing."

Spicer: Oh, that's great. What was happening in middle and high school during these years? Any subjects that really appealed to you in particular?

Colbert: I mean, I think a lot of what was happening was <laughs> I was nerding out with other computer geeks. I've always been interested in learning more, and school's been… and college after that was a great avenue to learn about a whole wide breadth of different subjects. But computers and technology in general was always something that really interested me. I think this notion, especially around

programming -- the ability to build a system that could to some degree exist on its own after you set it up and started it going -- was something really, really powerful. It always intrigued me very deeply.

Spicer: Did you have any heroes growing up, either technical or otherwise, people you admired? Let's put it that way.

Colbert: I think definitely a lot of the early computer pioneers. I mean, Bill Gates of Microsoft obviously is a big one. Steve Jobs, I think, for me, hadn't become that big. He would become a bigger force later in life with the advent of the iPhone and all the great stuff that he did there. Let me think.

<pause in thought>

Colbert: Yeah, that's a good question. No one else is coming to mind.

Spicer: That's okay, yeah. <laughs> We are--

Colbert: <overlapping conversation> I'm sure there'll be some others--

Spicer: It's a tough--

Colbert: -deep in the recesses, yeah.

Spicer: Yeah, it's a tough question. I mean, it could be a politician or a philosopher or anybody. It doesn't have to be a technical guy, but...

Colbert: So let me think. I remember at the time another area that had really fascinated me, really more in high school, was the philosophy of mind. How does consciousness arise? So one of the philosophers I really appreciated at the time, and still do, I suppose, is Daniel Dennett. So he's a philosopher, a professor at Tufts, I believe, or was, and he had this great book called "Consciousness Explained", which was really interesting. It talked about how you have this sort of more traditional neural circuitry, if you will, of the brain, and the consciousness was somewhat of an emergent property that created, in his terms, a virtual machine, not to be confused with <laughs> the VMware virtual machine. But this virtual machine that essentially leveraged this kind of older hardware, if you will, to create this newer sort of software abstraction. So it was interesting the way he thought about it, clearly very influenced by a lot of what was happening in the technology space, in the computer space. But to explain this thing that today we still don't have a great understanding of 30-odd years later, right? So I think Dan Dennett is someone that I never had a chance to meet, but always highly respected in terms of his creative thinking ability.

Spicer: Interesting. Well, that's a great example, yeah. Now, in high school, did you do any sports at all?

Colbert: Yeah. <laughs> So for the longest time, I was a big runner, and then I kind of got out of that for a while and then didn't get back into it until I came out here to Silicon Valley. But during high school, I was actually really into hockey. Now, this is weird for someone living in Oklahoma because we don't get-- I

mean, we get cold, but we didn't get that much snow. Certainly not that much ice or frozen ponds <laughs> or anything. So instead, what we did was roller hockey. So there was actually, surprisingly perhaps, a pretty large contingent --underground contingent -- of roller hockey folks in Oklahoma. I played with friends locally and then went to a number of different leagues and did a lot of that. So I was really into hockey during my high school years.

Spicer: Oh, that's really fun. Yeah, that speaks to my heart as well, of course, as a Canadian. <laughter> That's wonderful. Wow, very nice.

Colbert: Yeah. That was actually one of the things I was looking for in a college, was hockey. I mean, really more of the intramural. I wasn't planning on trying out for the varsity or any of that stuff, JV. But I wanted a good hockey program, a good computer science program, and then also kind of a smaller town. I didn't want to be necessarily in New York or anything like that, I'd be too distracted. <laughs> But yeah, hockey was kind of a core part of me at that time.

Spicer: Wow, that's really interesting. Now you ended up going to Brown?

Colbert: Yeah.

Spicer: So tell us about that decision. Did you apply to other colleges?

Colbert: I did.

Spicer: Which ones and give us the story of...

Colbert: Yeah, Brown was very formative for me in terms of my growth and evolution as a person. Both, I think, just purely personally as well as academically and from a career standpoint as well. So, as I mentioned, growing up in Oklahoma, first of all, I think I was somewhat… or not very exposed to the wider world, not super cosmopolitan. <laughs> So I had a lot to learn, which I think I had some understanding of. So as I mentioned, I was really looking forward to, or looking for a college with a strong computer science program because I knew I wanted to do something in the computer science realm. Not sure exactly what, but I wanted something with a good program. And then hockey, because I loved hockey. And then, as I said, kind of a smaller town. So I looked at a bunch of different places. I sort of looked through the emerging online <laughs> databases of colleges and a lot were in the Northeast, a lot in New England. So a lot of the Ivies, Brown, Yale, Harvard, Princeton, Cornell, but a few that weren't on there, Colgate. And then, interestingly, one over in Colorado, Colorado College, I think it was? So went and visited all these places, did kind of a New England tour.

I remember going to Brown and just setting foot on campus. There was something about the campus that really just spoke to me at a very deep level. I just had this felt sense of "This is the place." I remember going, really enjoying my conversations with professors there, with students, just kind of talking with people. Just the vibe was awesome, and particularly the computer science program, I really appreciated. The program was very hands-on. You do a lot of coding, or at least you did at the time. Maybe it's still the

case today. I'm not sure. But there it was very coding intensive. You built a lot of things, and that really spoke to me as well. So I just had this sense that immediately being on campus, I was like “This is the place for me. This is where I want to go.”

Spicer: It sounds like, and I don't want to put words in your mouth, but was it kind of a Renaissance vibe, the arts and sort of the--

Colbert: A little bit, yeah.

Spicer: -science and the humanities together [ph?]?

Colbert: No, you're exactly right. I think that's actually a very good point and maybe something I forgot to mention earlier, was that I did not want to go to a purely technical college or university. I wanted a liberal arts education, and wanted to… even though I knew I was going to be a technologist, I always wanted to have that broader purview. Because I do have a lot of interests. We talked about this, this philosophy of mind and <laughs> some of these things. I had taken a number of philosophy courses at Brown, actually, as well. But just to be exposed to this broader set of subjects. So that was a really important aspect for me. So the fact that Brown had both a really strong computer science and more broadly engineering, math, etc departments, as well as these really strong non-technical departments, humanities, etc, was another big reason that I decided to go there.

Spicer: Now I want to ask you, well, two things. One is there's a great quote by Steve Jobs who says “It's the marriage of science, the technology, and the humanities that really makes our hearts sing.” I love that quote.

Colbert: I think that's a beautiful quote.

Spicer: It's so-- well, I don't want to make it an ad for Apple, but it's very apparent, I think, in what they do.

Colbert: Yeah. Well, and I think-- and we can talk about this as the conversation progresses, but I do think that a lot of-- I look at my career, I've clearly been afforded a great set of opportunities, especially at VMware. But I think a lot of the reason I've been able to grow in my career as I have is because of this very multidisciplinary mindset and background that I have. Especially as you evolve in your career as a technologist that I think many of us, like myself, started as individual contributor engineer, but as you move higher in the ranks, it's more and more about interacting with people. It's about taking on and harnessing these non-technical skills. So I think this notion of blending technology with these non-technical areas is important throughout life, throughout your career. So I think I was really fortunate to have that intuitive sense that this was important and to really parlay that into my time at Brown.

Spicer: That's wonderful. We have a couple of curators here who went to Brown, and they love it, too. We made Andy van Dam a fellow--

Colbert: That's awesome, yeah.

Spicer: -a couple of years ago. So that was nice. Speaking of Andy, I think what I want to bring up [next] is how important undergraduate education is at Brown. I mean, the professors really take it seriously.

Colbert: They do.

Spicer: They're not just [doing] research in a little lab somewhere off by themselves. So do you want to comment on that at all, how good the professors are or what they were like?

Colbert: Yeah. I mean, in some ways, I don't think I knew how good I had it <laughs> because I didn't go to any other colleges. I didn't matriculate anywhere else. But in general, yes, one of the things I really did like about Brown was the engagement with the professors and the level of time and overall engagement and effort they put into the undergraduate population. It showed in a number of different ways.

Obviously, there were the classes themselves where the professors were very hands-on. They weren't just throwing it to one of their grad students and kind of barely showing up. No, they were developing the curriculum. They presented the material. Grad students would be involved, but usually as support, off hours, or these other sorts of things. So you had a lot of face time with the professors.

But the other part that I really liked about it was the teaching assistant program. So there you had not only-- sometimes you had graduate students, but most of the time it was undergrads who would TA these classes. The cool part about that was that it got you not only more face time with the professor, but it also really helped you to deepen your understanding of the course material. It's one thing to be able to take the class and pass it and get good grades, it's another thing to help to teach it because then your own understanding of that material has to be to the next level. You're developing technologies and courses and curricular questions. You're having to answer questions. So I remember I TA'd many different classes, and that was really huge for me as well, as I said, both to get the face time with the professor, but also then to better understand the material myself.

So I do think there were a lot of these unique aspects of Brown. I guess maybe a third thing I would say is that a lot of the research work we were able to be involved in as undergraduates as well. So I spent a lot of time in what we called Syslab, the systems lab, because I became a big systems guy, helping to do some of that academic research work as well. So I think the opportunities that we had, and at the time it was a fairly small department.

Spicer: This is the computer science department?

Colbert: Computer science at Brown. I mean, it's grown <laughs> tremendously since I left, where I think the CS graduates now make up at least a quarter of the graduating class, which is crazy because it was tiny back when I was there. In any case, the point being that it was a fairly small department, and everyone knew each other fairly well. So that was another really nice thing about it.

Spicer: Also, I believe you had a nice benefactor in Thomas Watson?

Colbert: Yes.

Spicer: The chairman of IBM <laughs> who went there…

Colbert: Yeah.

Spicer: -and subsidized-- or helped IBM fund some of the computing facilities there.

Colbert: Absolutely.

Spicer: Did you have any sort of breakthrough moments in your education, either in the computing side or where you really got excited about a topic?

Colbert: Yes, I absolutely did. I think one of the big breakthroughs for me was during our systems class, and-- during the operating systems class, I should say. So I think this is what really gave me clarity that I wanted to go into that sort of systems, operating systems area as something I wanted to do professionally. I remember very specifically what it was. So as I mentioned, Brown CS, very heavy on programming. You just do a lot of programming there. So one of the classes, CS-169 it was called at the time, was the heavy programming version of the operating systems class, where essentially you wrote not quite an entire operating system kernel, but the majority of one, and a lot of the critical functions. I remember this one function, it was called swtch.

Essentially, a swtch's job is to switch between threads or processes of applications. So this is a very foundational thing, and what's cool about-- you kind of write it and you're building it, there's this moment of understanding where I was like “This whole notion of a thread or a process, it's all an operating system abstraction.” The underlying processor has no knowledge of that. They have ways of helping you deal with it, but that abstraction has to be created by the operating system and handled and held by the operating system. It's kind of funny because if you just look at the code for switch, when you enter into that function, the swtch function, you're operating within the context of one thread or process, and by the time you are done with it, you're operating in the context of a different thread or process. The kind of magic happens in the middle.

That process for me of building that function was so foundational because it kind of was like “Oh, yeah,” gave me that insight that “Oh, we are creating these abstractions. We are doing this sort of thing at such a fundamental level.” It was really mind-blowing for me and kind of exciting for me to say “Hey, there are so many opportunities to create this abstraction, sort of to take a lot of that complexity and abstract it away.” I thought that was something that really shifted my thinking and made clear to me that “Hey, this is an area I want to continue to experience and experiment with and investigate as I move forward.”

Spicer: Right. Well, this sounds very much like, and I may be wrong here, but your first job at--

Colbert: Yeah <laughs>.

Spicer: -VMware, where you're dealing with multiple processes. I think we can make the jump now from Brown to VMware. Was that your first job, at VMware?

Colbert: It was, yeah. So my first job out of college was at VMware. I actually interned at VMware between my junior and senior year of college at Brown. What I worked on, both during my internship and then early on in my career at VMware, was actually the operating system kernel that we had for our ESX product. So for those that don't know, ESX, part of vSphere now, but back then it was its own separate product. So essentially what VMware did, obviously its claim to fame, initially, was the hypervisor, was virtualizing the Intel ‘x86 processor, which a lot of people didn't think could be virtualized. A processor was never designed for that.

Spicer: Now, let's take a break, just so we can bring along people who may not be experts in virtualization--

Colbert: Yeah, what is virtualization?

Spicer: Yeah, can you tell us what that is? Maybe just include a bit of history too, because I think it has some prehistory--

Colbert: It does.

Spicer: -that we could look at as well.

Colbert: Yeah, so virtualization, the basic idea, the way I've always described it to friends, non-technical friends and family, <laughs> is this notion of you're essentially running multiple small computers inside a bigger computer. Concretely what we're talking about is that we can fabricate all the different physical aspects of hardware in software and present them to an operating system as if that operating system’s running just on physical hardware. So it's a bit of an abstraction, if you will, and this is what we do. We faithfully replicate properties of hardware in software such that an operating system thinks it's running on actual hardware. So as you mention, virtualization was not new to VMware. VMware did not create the concept. Instead, virtualization had been around for decades before that. A lot of the older mainframes were designed for virtualization, so you could run multiple operating systems, instances of the operating system side by side, and each--

Spicer: Was that IBM's VM...?

Colbert: Yeah, some of the early IBM, and then I think there were some other ones as well. But basically, the idea was that you could have that level of separation. That way, if any of the individual operating systems had problems or issues, the other ones wouldn't be affected by it.

Spicer: From a cost point of view, because that obviously is a huge driver behind all of these things, what is the advantage to the person paying the bills and to the user?

Colbert: Yeah, so why do it? So getting back to VMware, what had happened was, in this time, this was kind of late ‘90s, early 2000s, Windows was by far the most dominant operating system out there in the data center world. Because of the design of Windows, Windows was originally designed more as a PC or end-user operating system. So really, it was only meant to be running one or two major applications at a time.

Now, in a server environment, the servers are obviously much bigger <laughs> physically and in terms of their components. So there was no reason they couldn't run multiple applications at a time. The problem was actually Windows itself wasn't really designed well for that. Different apps had different dependencies. They needed different libraries and different versions. Essentially what happened was trying to run two major applications that had these differing requirements was just too complex. So, what companies did was they just bought multiple servers, installed Windows on each, and ran one app per server. So it solved the complexity problem, but now what you had was a utilization problem, where you had all these servers sitting around, beefy, expensive servers. Maybe you're only using 5% of their capacity.

So we had this problem where you had a lot of servers, one app per operating system, per server, but very low utilization. So this presented the opportunity for virtualization to solve that problem, that you could have one big server, physical server, have multiple virtual machines, each running an operating system, and each of those operating systems would have one application, just like before. So the benefit would be that it would look like each app was by itself, which it was in that OS, and each OS was by itself in the hardware, which was, of course, not true. Virtualization abstracted that.

So VMware was able to dramatically reduce the cost for running a data center from having all these expensive duplicative servers with really low utilization to now having a smaller number of servers that were much more highly utilized because they had all these virtual machines running on them. The benefit to the user, to the admins, were that they could administer these Windows servers in the same way as before, that they could have one major app per Windows’s instance, the “DLL Hell,” they called it, and they could manage all that, that'd be fine. So they did not need to change their administrative processes. To users of those servers, they didn't notice that anything had changed. It was the same old Windows app <laughs> as they had before. So really it was benefits across the board, and I think this was the opportunity that the virtualization technology solved.

Spicer: Thanks for explaining that. That was really clearly explained. Thank you.

Colbert: Of course. <laughter> I've done it a number of times now.

Spicer: <laughter> I guess you are familiar with the technology! Just to drill down into the hardware here, how important were increasing hardware or CPU speeds to be able to pull this off?

Colbert: Yeah. So a couple more things on the background of this. Now, one of the things that VMware pioneered was virtualizing ‘x86 processors. Now, unlike the IBM mainframes from decades before, which were specifically designed to virtualize, ‘x86 was never designed to be virtualizable, and it was not known

at the time whether we could actually virtualize it, whether that was even possible in theory, or whether it could be done in practice, i.e. done with good performance. The early days of VMware were it'd take like 10 minutes to boot up Windows. <laughs> I mean, this was before it got released. The very, very early sort of beta versions. So that was the key solve, was that we were able to-- I say "We", I can't take-- <laughs> VMware was able to solve this, both from a practical standpoint, actually from a theoretical standpoint, and from a practical standpoint, that it had speed. Now, from the early days, even after the initial releases, you always had a performance penalty for that. This is back, again, in the early 2000s.

Spicer: Is that for the hypervisor?

Colbert: Yes. Because what we did was the hypervisor essentially did all that virtualization itself in software. So it had to sort of read through the operating system code, had to rewrite it a little bit, and then execute these rewritten pieces of code, and they were executed in such a way that the hypervisor could take back over if anything bad or anything else happened. So essentially, the hypervisor was doing all of its work in software, and not relying on any special processor capabilities.

Now, obviously, the processors were getting faster, and that was great, because we could allow the hypervisor to run faster, but there was still this hypervisor overhead, and we wanted to solve that. It's kind of funny, actually. One of the other historical asides I think a lot of people forget was at the time, in the early 2000s, a lot of pundits, journalists, etc, were predicting the death of Intel, because they said "Oh, VMware's coming, virtualization's coming. We're not going to need processors anymore," all these sorts of things. So it was looked at as a huge competitive threat for Intel, but we at VMware actually saw it very differently. We knew that there was a tremendous amount of innovation that we could do together with Intel.

So from the very early days, engineers at VMware worked with engineers at Intel to say "Hey, how can we come together and optimize both what we do in software for virtualization with what Intel's chips offer from a hardware standpoint for virtualization?" Now, today, the very advanced Intel processors have all sorts of great virtualization extensions that really take the load off of the hypervisor. So what we've seen is the evolution of the hypervisor, where before we did a technique called binary translation, but by the 2010s, that was no longer necessary. The processor really did most all of the virtualization at that point, and the hypervisor just coordinated some of that.

So what we've seen is because of these amazing advances in processor technology, virtualization overhead today is essentially zero. As a matter of fact, what's somewhat counterintuitive, the most recent, even like the past three, five years, performance test shows that in some cases, an app running on VMware vSphere can actually be faster than an app running on bare metal, on Linux.

Spicer: Now, just tell us what vSphere is? Is that your main part…

Colbert: Yeah, vSphere is our hypervisor. So running on the VMware hypervisor, you can actually have an app that's faster than running on bare metal.

Spicer: Amazing. How did this process of mutual alignment between VMware and, say, Intel, AMD happen to get the hooks for the hypervisor in there? Do you know how that happened?

Colbert: Yeah, well, I think there was clear incentive across both companies. Intel, obviously the standard in the data center world in the 2000s, and even today, and their processes were everywhere. VMware, still being an up-and-comer at the time, we wanted to make sure that we worked well on all hardware that we ran on, particularly on Intel, just because that was the vast majority of the processors that VMware's hypervisor ran on.

So there was incentive for both companies to come together, and it was one of those situations where the engineers got together and really we walked Intel through what we were doing, how our hypervisor worked, and then also where were the pain points. What do we need in an ideal world from the Intel processor? What would we love to have? You have to understand that on the VMware side, we have some super low-level hardcore engineers that would go down and know the details of the Intel processor.

Actually, as a fun aside, every processor, Intel, AMD, ARM, all those processors, they have bugs in them. They have specifications about how they should work, and then slight deviations from that in terms of how they work in the real world. What's interesting is that VMware has to faithfully reproduce the bugs of processors in order for virtualized applications to run properly. Because believe it or not, these apps sometimes rely on bugs in processors. So we knew very precisely how these processors worked, the ins and outs of them. We knew exactly how long different instructions took, how many cycles it took to clear cache, or if we had a misprediction, a branch misprediction, these sorts of things. We had all these numbers dialed in, and all of those fundamentally impacted the performance of our hypervisor.

So we were extraordinarily hands-on from a technology perspective with these processors that we ran on. So we'd sit down with the Intel engineers and say “Okay, here's what we're doing. Here's what we're seeing, too many cycles.” Or “We want to optimize this,” or “We'd love to have an instruction to do this other thing,” so on and so forth. So really it was sitting down, engineer to engineer, and plotting out what needed to happen to get to this point that we're at today.

Spicer: Amazing. Did you do that with AMD and ARM as well?

Colbert: Yes. VMware, again, a huge partner with Intel, but obviously worked with ARM, and then--excuse me, worked with AMD, <laughs> and then further on worked with ARM. ARM was one of the interesting ones. Obviously, we had a huge market with ‘x86, that was the market we targeted from the beginning. ARM was one of those up-and-coming processors that we're always sort of watching and waiting. In fact, it wasn't until my time, I actually incubated for a while the ARM team within my organization when I was CTO for our infrastructure business unit. We were always sort of wondering “Do we officially support ARM? Do we not?” It was not really did we want to or not, it's like “What's the market opportunity?”

For us, because there was sort of a chicken-and-egg problem there, it's like you go support ARM, but there's not enough hardware and ecosystem and so forth to support ARM in the data center. But what

actually tipped the scales for us was the SmartNIC which came out, and there's a number of vendors who produce SmartNICs. A SmartNIC essentially is a NIC, which is a network controller, [that] speaks to the network. What they started to do was put general-purpose CPUs, and really SoCs [Systems-On-Chips] -- so whole small computers <laughs> -- onto that NIC. What that meant was that we could then run a copy of our hypervisor on top of or on the NIC. So these NICs oftentimes would be ARM-based, and so as we started to see more and more popularity of SmartNICs, we decided to actually productize, finally, our hypervisor for ARM. So that took us almost 20 years <laughs> to do as a company, but we eventually got there… yeah, again, working very closely with ARM and many of the different manufacturers in the ARM ecosystem to get that done.

Spicer: Now, since you mentioned networking, I'm curious, VMware also produces a networking virtualization product…

Colbert: Yeah.

Spicer: -as well? Can you explain what that is and how it works?

Colbert: So let me take you through the history of this expanding scope of virtualization. <laughs> So we started off with compute virtualization, and typically speaking, if I use, or almost anyone, if they use the term “Virtualization” without any additional qualifier, they're almost definitely speaking to compute virtualization. So if you just say “Virtualization”, you're talking about processor and memory virtualization. Now, that's only one part of the computer. Any computer’s got multiple different types of I/O, you've got network, you've got storage, you've got sometimes GPUs.

There's many different things that you can virtualize. So obviously our initial success as VMware was with compute virtualization, and we got great feedback from customers, as I mentioned. This notion of being able to consolidate all these different physical servers from many that had low utilization to a smaller number that had a much greater utilization, that was awesome. But what we heard from customers was “Okay, love it, you're doing great stuff for us. But my problem is that I can now create this virtual machine, this software construct, that takes like five seconds. But in order to get storage, I got to go call my storage person and have them create a LUN for me, or file a ticket to go do that. That might take a few days or a few weeks or however long. I got to go and talk to my networking person and get a new network, VLAN, created for me, or whatever else.” Usually that was, again, a ticketing system or some sort of manual process.

So in order to get a new application provisioned, we could create a virtual machine in a few seconds, but all these other tasks would end up-- it means it would take weeks for that app to get provisioned. So that gave us the idea to say, “Hey, can we take this notion of virtualization and extend it to these other domains?” To expand it from just compute to cover storage, networking, and other things, like eventually GPUs, for instance. So that was really the second lap of the journey for VMWare. The first lap was core compute virtualization. The second lap was really expanding that to virtualize the entire data center.

The vision we had for that was called the software-defined data center. So we did that in a few different ways. So on the storage side, we actually built a homegrown technology that came to be called vSAN, virtual SAN, and this was a way of essentially virtualizing storage across a cluster of vSphere hosts, again, vSphere is our hypervisor, and to take all these local disks and make them look like they actually are available to any host in the cluster. So essentially, it's a distributed storage concept. There's a lot of really hard tech that went in there.

Similarly speaking, on the networking side, we created this technology we call NSX, and that came from an acquisition we did called Nicira, as well as some previous acquisitions, as well as some homegrown, kind of put together a bunch of different pieces. Now, the networking one was fairly powerful because this notion of network virtualization, again, people didn't know if that could be done or how it could be done, or could it be done in a way that was actually performant and could be used in an enterprise environment. It took quite a while to figure that one out. But eventually, when we got NSX out there, that became really powerful technology, because what it meant was that we could actually have this virtual network managed by NSX overlaid on top of the underlying network in whatever configuration they had. You didn't have to necessarily buy new hardware. You didn't have to necessarily reconfigure the underlying network.

So similar to what we did with compute virtualization, where you didn't need to change your app, you didn't need to change your OS, you didn't need to change your hardware, we now had the same property for networking. So bringing all these together, vSphere, our compute hypervisor, vSAN for storage, and NSX for networking, we have what we call the software-defined data center. That was really the second lap of the journey for VMware, where we endeavored to virtualize the entire data center, allowing customers to much more quickly stand up new applications, to not have to file tickets and send a carrier pigeon somewhere to do whatever, but instead just to call an API and everything was created for you instantaneously.

Spicer: That's incredible. I don't know how free you are to say who some of your biggest customers were, but if you are, could you name some of them?

Colbert: Yeah, absolutely. So I mean, I can speak with some generalities, I'll say. So essentially, we had <laughs> all the Fortune 100, all of the Fortune 500, most all the Fortune 1000. We went across industries. One of the beauties of the VMware technology was that it was so general purpose. It didn't matter what sector of the economy, what industry vertical you were in, you needed the tech that we had, because it goes back to conversations over the past two decades. Every company’s becoming a software company. Every company needs this sort of technical know-how, and so everyone was building out their data centers.

So many, many large banks, folks like JPMorgan Chase, for instance, huge customers. We had folks in healthcare. Many different large and small clinics and hospitals around the world using our software. Folks in manufacturing. The list sort of goes on and on. So it's really across the board. I think because of the early success of vSphere and just that simple value proposition of “Hey, we can dramatically reduce the number of servers you need. We can reduce your data center footprint. We can reduce the amount of

money you spend on cooling and power and all these things for your data center." That was just such a no-brainer that it really grew like wildfire across all these different industries.

Spicer: I'm going to assume once in a while somebody's application didn't run. Maybe it happened more at the beginning and--

Colbert: Yeah.

Spicer: -it's basically down to zero now?

Colbert: Yeah, essentially. So what happened was, like any new technology, there was always a little bit of concern about it, and especially a little bit of concern around running a business-critical application on it. So the general trend you see for most of these new technologies is that you start out with some lesser criticality application, sort of kick the tires on that one, and then as that proves good, then you move up the stack, so to speak. So a lot of the early applications that were moved on to vSphere were things like file and print servers. These weren't going to be the mainstay, giant databases, all this sort of stuff, or maybe customer-facing applications. These are more the back-end stuff, maybe your HR application, these other sorts of things. Not like they went down very often, but if it did go down, okay, it wasn't the biggest deal in the world.

So, a lot of our customers wanted to prove to themselves that "Hey, this thing will work with my less critical applications, then we can move it to the more critical ones." So what we saw was just that natural progression. Now, in addition, there were real technical issues with some of these high-performance applications at the beginning. As I mentioned, when vSphere, the hypervisor, was still using binary translation, there was a performance overhead, and it varied. You might have 10% overhead in some cases, many cases less, maybe sometimes in extreme cases more. But obviously, if you have a mission-critical database, you probably don't want to mess around with that one yet.

However, as we started working with Intel and all the different chip companies and started optimizing, and we're doing a lot of things within our hypervisor. Getting to greater and greater performance, more latency-sensitive applications. Folks on the telco side were always somewhat hesitant to move to it in the first few years. But as we did that, as we supported larger and larger memory sizes, more and more virtual CPUs, eventually we were able to get to the point where we could start virtualizing databases and things that were just extraordinarily performance-intensive.

So it took the better part of, I'd say, 10 to 15 years, but eventually we got to the point where you started to see many companies say "You know what? I'm going virtualization first." Before it was "Oh, I need a reason to go to be a virtualized app." Now you need a reason not to be virtualized, a reason to go into bare metal. But we started seeing that, again, across many of our customers in, let's say, 2010 to the 2015 timeframe. Really, after 2015, it was kind of standardized everywhere. So there were a lot of technical hurdles to break down, but also a lot of non-technical ones. A lot of people just saying "Oh, I heard about virtualization back in 2000, and you guys had a lot of overhead at the time." It's like, yeah, we did, but it's 15 years later and we don't anymore. So a lot of times it was that sort of updating people's

priors, if you will, kind of re-educating a little bit to say “Hey, the world's different, things have changed, technology has improved dramatically, and now we can absolutely adopt these mission-critical apps.” What you see today is that it's by far the default to have these mission-critical applications running on vSphere, hypervisor.

Spicer: Right, and I think I asked Steve Herrod this, and since you mentioned J.P. Morgan, were they not your first largest sale? Or one of the...?

Colbert: They may have been. I don't know if they were actually the first, but they were definitely very early, very, very large.

Spicer: Or a really key sale for you?

Colbert: Yeah, a lot of the financial institutions were early customers. Financial institutions in general-- <laughs> well, in some ways I think people look at them as a little bit-- maybe a bit more conservative. We found them to be extraordinarily leaning in to new technology, looking at how they can get any sort of edge, looking for greater efficiency, greater automation. Obviously with the vSphere hypervisor, it did lead to a lot of automation that now what used to be a physical act you could just do with an API. So we've always been really close with our financial customers, just because they continually push us on the cutting edge in how to be better.

Spicer: It's interesting because, I mean, there's real money involved, and for them to take you on is a real sign of confidence.

Colbert: Yeah, and it's one of those interesting things. As I mentioned, usually they start with less mission-critical or non-mission-critical apps, but oftentimes they would eventually move their trading applications, things that have very specific latency requirements. So our engineers, remember, we'd send out our top engineers to work with them and say “Okay, here's how you tune this setting and do that to drop the latency even lower, so that you can get the best performance in terms of trading responsiveness and everything else you need.” So they were always really great in terms of pushing the envelope and pushing us to be better and really helping us set the direction for how to evolve the technology.

Spicer: Another thing that helps you push the envelope is to have competitors.

Colbert: Yeah <laughter>.

Spicer: Did you have any, or do you have any still?

Colbert: Oh, absolutely. The nature of the competitive landscape has changed dramatically over the years. So if you look back to the early days, VMware was focused on the PC market. We had VMware Workstation. That was the initial product. Really, it was targeted toward technical users. Eventually we got more toward, hey, the everyday mom and pop, <laughs> Joe Schmo, non-technical. But the beginning was much more geared toward technical people, and we did have a number of competitors in that space.

I think one was called Connectix, if I'm not mistaken? In any case, one of the interesting things early on was that competitor product, we felt, was infringing on some of our patents.

So there was some lawsuits. I wasn't close to it, but this is what I understand, some lawsuits that we initiated against them. But then what happened was that Microsoft bought this competitor, <laughs> and so now we're in the position where we're in a lawsuit with Microsoft, which obviously is never good, being very small like we were. I think we started to see success in the server market at that point, and that's what actually led us to-- the potential lawsuit with Microsoft and some of the fallout there actually led-- was one of the considerations that led to VMware's acquisition by EMC in 2003, 2004 timeframe. So that kind of set the tone a little bit.

Now, after that, as we started getting deeper into the server market, we then saw Microsoft come out with Hyper-V, and that was their sort of answer to virtualization. As I mentioned, the biggest operating system workload we had on VMware at the time was Windows. We were seeing more and more Linux. Obviously, Linux was growing in the marketplace, but by far Windows was the most common operating system. So we were really worried when Microsoft entered into the market with Hyper-V because now we were competing directly against Microsoft who could build Hyper-V into their own operating system, which was the most common one that we were using.

Spicer: Like they did with web browsers.

Colbert: Exactly. We've seen this playbook before. So that was a huge strategic threat to us. I think the good news was the fact that we had such a strong head start in terms of technology, in terms of the performance, that we were able to argue that “Hey, yeah, Hyper-V may be quote, unquote ‘Free’, but when you're paying for VMware, here's all the benefits that you're getting.” I also think, longer term, what we created with VMware was a really powerful management ecosystem around virtualization.

So it wasn't just the virtualization tech, it was the whole management environment. You started to see companies adopt that whole management environment. That's another thing that I think really helped protect us against what was happening in the Hyper-V space. Now, there's some other areas besides compute virtualization. On the storage virtualization side, we had a number of different competitors there. We haven't talked at all about end-user computing, what we did for virtual desktops. We had Citrix, was a very large competitor there. So really, we're battling competitors on multiple fronts, and as we moved into VMware's third lap of the journey, if you will, as we went into cloud, we saw even more.

Obviously, public cloud came out, and there was a whole public cloud versus on-prem sort of competition. We were a part of that on-prem data center market, which meant the public cloud sort of presented a challenge to us. At first, we did try to-- and this is probably one of the largest, I think, shifts that we had to do as a company. At first, we did try to compete directly against it and did all the normal sort of things you do, <laughs> creating some of the… bringing up a lot of the issues at the time with public cloud. It wasn't as resilient as on-prem data center, all these different things. But obviously, public cloud continued to evolve and improve. So I think one of the big strategic shifts that we did as a company was say “Well, rather than try to fight public cloud, why don't we join forces with them?”

So I think one of the biggest announcements we ever made was a partnership with AWS, that happened back in 2016, where we said “Instead of trying to take workloads back from AWS, let's move VMware workloads into AWS and support customers wanting to go there.” So we actually partnered with AWS to get our hypervisor into their cloud environments, and that was a huge benefit for customers, and then we also did that for all the other hyperscalers, with Azure and Google, IBM, everybody else following suit.

Spicer: So all of these public cloud places are using VMware?

Colbert: Well, no. So, let's be very clear. All of them have their own hypervisors--

Spicer: Oh, I see.

Colbert: -that they've built themselves, but they have a specific service that uses the VMware hypervisor. So they do both, actually.

Spicer: It's an option?

Colbert: Yes.

Spicer: Yeah, I got it. Oh, that's great. Wow. You did mention the EMC purchase, and then, of course, there was the Broadcom one that just happened. Can you walk us through those two acquisitions and give us your thoughts on that?

Colbert: Yeah. So VMware started as a company back in ‘98, and--

Spicer: Yeah, sorry to interrupt. We didn't really talk about the founding with Diane. Could we start your answer again with the founding?

Colbert: Yeah. So let's rewind the clock. So VMware was founded back in ‘98. So, first, a couple of things. So it was founded by Diane Greene, who was the initial CEO, her husband Mendel, who was a chief scientist and also a professor at Stanford.

Spicer: What is his last name?

Colbert: Mendel Rosenblum.

Spicer: Thank you.

Colbert: Thank you. <laughs> I just know him as Mendel, but yeah, I know. <laughs> And some of his grad students from Stanford. So there was Scott Devine and Ed Wong and Ed Bugnion. I think it was those five folks. Am I missing anyone? I don't think so. Anyway, so five founders. Now, Diane had previously started another company called VXtreme, which I think she might have even sold to Microsoft. So she obviously had business experience, startup experience, and what was happening was Mendel

and his team were working on-- his students were working on this virtualization concept, “How can we virtualize commodity systems which weren't designed to be virtualized?” They'd done enough research where they believed “Hey, this was possible.” So they started VMware as somewhat of an extension of that Stanford research work to take these ideas they had and try to put them into practice, to actually try to virtualize the ‘x86 processor. The core goal of the company early on was that they wanted to create this virtualization product. They wanted to be able to do it on unmodified commodity hardware with an unmodified operating system, an unmodified application. Now, you have to understand that there were many different attempts at this in the past, and almost all attempts require some sort of modification somewhere. You had to assume certain things about the hardware or make changes to the operating system to do something special to sort of work with you.

But no, because the founders had this vision to really dominate the data center marketplace, they knew that they couldn't rely on applications being changed. There are hundreds of thousands of apps out there. They're not going to go change them all. It's impossible. You've got to be able to work with what's out there, with what's there today, completely unmodified. As I said, before, it was unknown whether it was even theoretically possible to do this in any way, correctly do it, and B, whether it was possible to do it with any level of performance. So that was really the first few years of VMware's existence. Now, while the vision was always to get to the server market, to get into the data center, they also knew the data center is hard to get to if you have a completely unproven product.

So let's start off with hobbyists, technology enthusiasts, individuals. So that's where the strategic decision came to focus on those folks with the product VMware Workstation, which was the first instantiation of the VMware hypervisor. So that came out in either late ‘98 or ‘99 and kind of took the world by storm. People were kind of shocked that this sort of thing could happen. You could see multiple Windows machines or Windows and Linux VM running side by side on your laptop or on your desktop. So it got a lot of fame and visibility for the company early on, and they parlayed that into two different server products. So they were called GSX Server and ESX Server.

Technical details don't really matter too much. GSX wasn't really long for this world. <laughs> It didn't have really a good place. So ESX was the one that went forward, and that is the server product that everybody’s using today when they use VMware. So VMware vSphere, a component of that is ESX, which started out as ESX Server back in, I think, 2001 timeframe. So that was really the early success of VMware. Now, as I mentioned, VMware had a lot of success, growing very rapidly, making good inroads into the data center market. The workstation product was also heavily successful, seeing it used not just by individuals, but also by individuals within companies, and companies kind of leveraging the workstation broadly within their developer base.

So it was at that time that there was a lawsuit happening with Connectix. I think it was Connectix? Again, I forget all the <laughs> company names now. They got acquired by Microsoft, that put additional pressure on VMware to say “Hey, there's now a legal threat here, potentially, from some of these lawsuits to look for a suitor, to look for a bigger company to be a part of.” That was just one of the many different-- I wasn't personally a part of those conversations. I was <laughs> a junior employee at the time. But from what I understand, that was one of the parts of the overall consideration for the acquisition by EMC. I think at the

time, VMware was still growing strong, and what Joe Tucci promised or kind of set out for Diane Greene was “Hey, we want VMware to be successful. We don't really want it to become part of EMC completely. We want it to be its own independent company. We want you guys--”

Spicer: Can you tell us who Joe Tucci is?

Colbert: Yeah, Joe Tucci was the CEO for EMC at the time of the acquisition. Joe and Diane talked, and I think got to that level of understanding. It was really beneficial, overall. We got more access to resources. We got the sort of protection that EMC could provide against some of these lawsuits, frivolous or not. It really just allowed us to focus on growing the business, and that's exactly what we did. So it was many years of that, living as part of EMC, but it really was a separate organization. VMware actually then spun out to go public a few years after that, maybe I think it was 2009, if I'm not mistaken? We were still majority owned by EMC, but now a public company.

In 2014, 2015, EMC was then purchased by Dell, <laughs> which meant that now Dell had a controlling stake in VMware. So VMware's been through a number of these sort of trials and tribulations, <laughs> if you will. Most recently, VMware got spun out and became a fully kind of independent company outside of Dell and EMC, and that was just back in 2021. It was at that point that Broadcom made an offer to acquire VMware in May of ‘22. That deal just then closed back in November of ‘23. So VMware's been through quite a number of changing hands, if you will, in terms of different owners, starting with EMC, moving into Dell, and then to Broadcom.

Spicer: Amazing. That's quite a journey.

Colbert: Yeah, <laughs> a lot of change.

Spicer: One of your jobs has been to evangelize for VMware. I want to understand the role, how important that was in expanding once you got purchased by EMC, and so you could focus on growing the business, as you say. How important was that, going out to trade shows and conferences and all that extensive speaking?

Colbert: Yeah, so the evangelism was a key part. Really, there are multiple phases to it, particularly over my personal career. The first phase, in our initial growth, the first lap of the journey on compute virtualization, the big focus there was really trying to educate folks on how to get the most out of vSphere. A lot of my job at the time as an engineer was to explain about how a lot of these vSphere things work, kind of take a lot of the complexity, take all the different bells and whistles and nerd knobs, things we put out there and kind of boil that down and make it easy for people to understand. So I did a lot of those presentations. I went to a lot of what we call our VMUG, the VMware User Group conferences. VMUG's now an independent organization. They have conferences all over the world, very small ones of just maybe 5 or 10 people, to huge ones of 2,000 people. So I'd go to these things and talk about VMware technology, and it was awesome to see all the practitioners there, the admins, people who really were building their careers around VMware.

I think a key aspect there is to make sure that they are empowered to continue growing their career, to continue evangelizing on our behalf, which is really about how the word gets out there. So I viewed a lot of my role at that time as kind of like a teach the teachers <laughs> sort of thing. I was teaching them, but all the times they would go out and teach their peers or counterparts or friends or contacts, colleagues, whoever else. So a lot of that was sort of the technical education piece that I did. As we started getting into lap two and lap three of our journey, going from just compute virtualization to the entire software-defined data center and then on to cloud, a lot of that, my evangelism sort of changed. It wasn't so much technical education, but really it was trying to make sure people knew the breadth of what VMware actually did.

The challenge was that most people still looked at VMware circa 2003, 2004 as like “Oh, you guys do data center compute virtualization.” I'm like “We do, but we do a lot of other things too.” And really helping to make sure that they understood the breadth of what we do. They might not realize that “Oh, you guys do compute--” or, excuse me, “Storage virtualization and network virtualization now.” “Oh, you can also do virtual desktop management, mobile device management.” Or in more recent times, “Oh, you do software-defined WAN,” and a lot of these other things. So the list sort of went on and on. So a lot of it was continually educating folks on the expanding breadth of the product portfolio of the company.

Spicer: Right, and maybe focusing more on managerial arguments versus technical ones?

Colbert: That's a really good point. Yeah, the value prop for VMware early on was absolutely a technical one, and it was one that was driven by technical folks, people that could understand the technical rationale. Obviously, there was a huge business value there, this huge, huge ROI on-- the return on investment in VMware was like three to six months, and your total cost of ownership would go down dramatically by adopting VMware. Yeah, it was huge. But it was a very niche thing, and what was interesting, one of our… Carl Eschenbach, former head of sales, former chief operating officer, now CEO over at Workday, one of his quotes was like “VMware was the least disruptive disruptive technology ever.”

The reason he said that was because in the early days, you look at the technology, the hypervisor, and you look at the workflow that IT had. To provision an application, you'd have some requests come in from an app team saying “Hey, here's my app. Here's what I need.” Typically, before VMware, you had to go find a server or order a server, that would take five months or whatever, and then go get the network and the storage and coordinate all these different people. You had to talk to the operating system person and do this, that, and the other. It would take months and months and months. What VMware did originally was that middle piece on the server. Okay, you don't need to wait five months for a physical server, you can get a virtual server in three seconds. Great. But you still have all these other months and months of process. So in some ways, it was hugely disruptive because we could automate that.

But in some ways, it wasn't disruptive because that end-to-end cycle still took many months. So initially, we started out talking to that administrator who owned that one piece of this very long process. It was a very technical sale, very sort of technical focus, but very good business results. However, you talk to the CEO, the CIO, the COO, and they still saw this multi-month process. They're like “Oh, you took out a little bit of it, but I still got this giant thing here.” Not to mention “How is it affecting the rest of my business?” So

there was a big evolution that the industry went through as software “ate the world” and all these companies became tech companies, and folks like the CEO, the CFO, etc, had to become more technology aware.

VMware was very much in the background for a lot of that. So a lot of the big evolution was trying to put us more in the foreground to say “Hey, what we're doing is actually very pertinent to all the goals that you have. In fact, your ability to leverage our technology successfully will directly result in the success of some of these other goals, business goals that you have.” So there was a big change in sort of not necessarily who our customer was or who our user was, but who we need to talk to in order to make that sale happen. So that was a big shift as well and then kind of an evolution for us as a company.

Spicer: Right, and I guess the product got more expensive and bigger as well, <laughs> right?

Colbert: Yeah, I mean, vSphere has always stayed fairly well priced. What we started to do was add on a lot more stuff, that people needed not just vSphere, but all the other things surrounding it. That also, I think, created some complexity. There's just more components in there. There's more moving parts. It became a bigger sale because you had to make, as a customer, a bigger commitment to this broader set of functionality.

Spicer: So I have a couple of questions. One is, how quickly is VMware changing today? Secondly, you mentioned earlier virtualizing GPUs. So I'd love to hear your thoughts on that?

Colbert: Sure. Well, so let's see. So VMware’s been under a lot of change in the last few years. Obviously, most recently with Broadcom, just in the last few months. I'm not there any longer. <laughs> I left as part of the acquisition, as part of the changeover in the acquisition. But even before that, under our previous CEO, Raghu Raghuram, who was a longtime VMware person-- actually, I think he and I started almost on the same day. We'd both been there for a bit over 20 years. So he became CEO a few years back, and he was really directing some of the fundamental changes.

So let me talk about a little bit of my time as CTO there and some of the big changes we were trying to drive. So I took over as CTO in August, September of 2021 and was CTO for the company for about a bit over two years. My remit was a lot on transformation of the company. So as CTO, I owned a lot of our innovation efforts, our ESG efforts, as well as a lot of the common services and capabilities for engineering. A big part of what we needed to do was transform as engineering. Historically, VMware was a shrink-wrapped software company selling shrink-wrapped software. So the way you operated as an engineering organization, the way we operated, was very specific to that, that you're producing releases maybe once a year, sometimes once every 18 months, maybe some smaller ones in the middle. That you're actually giving software and having these customers operate it.

What we've been transforming to is really more of a cloud company, that you're delivering the software as a service. So there was some kind of foundational transformations that as an organization you have to make. It's not just technological, but it's also mental, it's cultural, it's procedural. So that was a lot of the big transformations that we were driving, was kind of these internal ones around how we operated as a

software engineering organization. So there's some pretty foundational sort of shifts that we were driving there. Moreover, our product portfolio was shifting. We were moving away from these perpetual licenses, which had dominated the industry for many years, to subscription license revenue, which is really where the industry's going, where Wall Street expects us as a company to go, as well as continuing to diversify our portfolio.

Obviously, vSphere and what we're doing on-prem still make up a big part of our revenue stream, but we're expanding that into the cloud. We're also diversifying, going into modern applications and things like security and different areas that we hadn't done traditionally. So these are a lot of the big shifts that were underway. I think what we're seeing now with the acquisition of Broadcom is in some ways a bit of a refocus and kind of getting back to basics. Whereas over the past few years, there's kind of an expansionist phase looking at new markets in different areas. What I see happening now is kind of a refocus on some of the core areas of VMware and making some really hard decisions about what to do and what not to do. So I do think there's going to be a lot more change coming for VMware. But in many ways, I think it's getting back to the core of what made us successful and really looking to expand upon that.

Spicer: Well, that's great. How about on the personal level? We haven't spoken much, and I don't want to pry too much.

Colbert: No, that’s okay.

Spicer: We all have lives while we have careers. Is there anything you want to mention about your personal life?

Colbert: Yeah, sure. I mean, or just maybe my personal growth. Yeah, so I mean, while watching everything that's happened at VMware, and there's been a lot of changes sort of at the company level and so forth, I was also really fortunate to grow up at VMware. I interned at VMware. I started there full-time out of college. It's the only full-time job I've ever had in my life. <laughs> So it's been very important for me from that perspective, and I've grown a lot as a person as well.

I started off as an individual contributor engineer, ended up as the CTO for the company with a team of 2,300 folks. I've learned a lot about being a person, <laughs> about being a leader, and it kind of goes back to what I mentioned earlier. When I first graduated from Brown, was first working at VMware, it was all sort of technology, and that was the big focus. Technology problems, how do we solve them? How do we get through them? But what I started realizing was that, as I was growing through the ranks of the engineering ranks, is that there's a lot of these non-technical problems that exist as well. I realized that I actually have a knack, both, I kind of think, a natural ability as well as ability to learn how to navigate these sort of non-technical-- a lot of people challenges. We talked about evangelism, and I think a lot of my strengths are actually on the evangelism side, not just evangelizing VMware to customers, but evangelizing ideas in general, even internally. I noticed that in my career as an engineer, I was very good at crisply boiling down a complex idea to something simple that could be crisply articulated to someone where they could get the sort of gist of it. So that's one of the things I've actually pushed for a lot during

my career, is helping to educate other engineers, and really anyone who cares to <laughs> be educated or to learn about these non-technical things.

One of the examples, one of the things that I started within VMware we called Pitchathon. So I noticed that during hackathons, teams work for 24, 48 hours, they're hacking out code, and at the very end, they pull together a presentation to present to the judges, and the judges then pick who's the winner there. My reflection on that was you spend all this time doing the hacking stuff, but at the end of the day, to some degree, the pitch that you give at the end, that five minutes, whatever you got, that's going to decide whether you win or not, irrespective of how good your prototype is that you built.

Now, I know most people do it for fun and whatnot, but it kind of was like no one really focused on that pitch. No one focused on the actual presentation. People assumed that the tech would speak for itself. So we sort of talked about it, we're like “Hey, let's turn this around. Let's just make kind of like a hackathon, but make it about the pitch.” So we called it a Pitchathon, because that's what it was about. It was just about <laughs> the presentation. You could do hacking before if you want, but you don't have to, because it's really about the pitch. You could pitch on anything you want. It doesn't matter.

So we brought in various people. We brought in this person, she was, or is, storytelling and really helping engineers to do storytelling. What is the story you're trying to tell in this five-minute pitch you got? How do you articulate concisely the points that you want to get across? We went into things like neurobiology of that, because engineers oftentimes don't like some of these soft skills because they think they're too whatever. But the reality is that we are kind of machines in some way. We have these neurobiological things that affect a lot of how we perceive what we see and arguments that we hear. The more you know about that, the more you can sort of leverage that and exploit that.

So anyway, these things like Pitchathon were about trying to educate engineers to expand their horizon a bit, to see these other sort of aspects. I think any sort of leader sort of needs that. So I think for me, one of the things that I've really been fortunate to gain is that perspective and insight about how to be a good leader. Yeah, there are these technical things, but there's all these non-technical things as well, even in the engineering domain, that you have to be good at as you grow in your career.

Spicer: That's wonderful. Is there anything else you'd like to leave us with today?

Colbert: I think that I really love the culture of VMware. I guess maybe a couple. Let me take a step back. A couple of things. So I spent 20 years at VMware, and I've seen good, the bad, and the ugly, so to speak. I think there's a tremendous amount of good. I think VMware is a great company, great culture, and it's a fantastic place to work. I've benefited tremendously from my time there, and so thankful I've had the opportunity to work there. But I think that one of the things, especially over the past couple of years, is that I've taken sort of a hard look at the company, at ourselves, looked at what can we change, what can we do better?

A lot of my time as CEO-- or excuse me, <laughs> CTO, I was trying to challenge some of these notions. I think there are important lessons here for anyone at any company, that what gets you to a certain point

won't necessarily get you to the next point, and that you've got to question and sort of challenge a lot of these assumptions. I think in some ways, or we look at VMware, while we had tremendous success with vSphere, huge success, in some ways that success got in the way of us achieving future success, because we relied too much on what made us successful there to try and achieve success in the future.

One very concrete thing, for instance, was, as I said, the culture was awesome at VMware, very supportive, very communal sort of culture. But it was also one of those things where you couldn't really take a hard decision, make a hard decision, and stick to it. It was kind of a decision by committee or by consensus. So I think that really held us back from achieving more. It was good maybe in the early days of navigating some of these things, but kind of held us back later on as we needed to make some hard calls on where do we invest, where do we not invest.

So I think the final thing I would say is that while VMware is a great company, at the same time I think, being hard on myself, I and possibly other leaders could've done better at challenging some of these cultural aspects that made us so successful earlier on to change them to set us up for better success in the future. So I think it's one of those things where any time you have success, you're fortunate to have it, you also have to be looking at how do you change and evolve to tackle the next thing.

Spicer: Yes. In essence, isn't that the so-called Innovator's Dilemma?

Colbert: It's absolutely the Innovator's Dilemma. Yeah. Again, I think we did some things really well and navigated certain aspects of that, and other things I think we could've done a lot better. So it's one of those things where I benefited hugely, and oftentimes I feel like you learn best through your mistakes, <laughs> and being able to be open and looking back and saying "You know what, we could've done that thing better. We could've done this thing differently." I think some of these cultural aspects especially are where I think we could've challenged that and changed that a bit more to have been more successful.

Spicer: Well, you didn't do too bad.

Colbert: Well, thank you.

Spicer: Well done. Thank you so much, Kit, for joining us today.

Colbert: Well, thank you for having me.

Spicer: Absolutely. Thank you.

END OF THE INTERVIEW

