



Oral History of Steve Herrod

Interviewed by:
Dag Spicer

Recorded October 13, 2023

CHM Reference number: 2023.0191

© 2023 Computer History Museum

Spicer: Okay, today's November 13th, 2023. We're here with Steve Herrod, who is a computing pioneer who has done many interesting things in the area of virtualization, and also in venture capital and finance. So, welcome, Steve.

Herrod: Thanks, Dag. This is fun to be here.

Spicer: I'm really glad you could join us. I wanted to start out knowing a bit about yourself personally, like your early years, what was school like? Did you enjoy it? What did your parents do, for example, and were they an influence? Those kinds of early formative experiences would be great to hear about.

Herrod: Yeah, so I guess [I came from] maybe a little bit of a non-traditional background. I was born in Louisiana. My mom was actually from the <laughs> swampland of Southwest Louisiana, Cajun Country. So I was born in the South and moved all around. My dad was literally a rocket scientist. He had his PhD in physics, and we went to White Sands Missile Range, which is near El Paso, Texas. We were at NASA in Huntsville, Alabama. So all kinds of fun, science-y jobs. My mom was actually a family counselor at a church, and so I had a PhD in physics dad, and a family counselor mom, which I think ended up being a really nice mix for engineering management. You have to have a little bit of both EQ and IQ, perhaps, <laughs> or something like that.

Spicer: Yes, exactly. The hard and soft skills, for sure. So your dad was-- tell me a bit more about him-- and then we'll get to your mom -- about the kind of work he did? What was he doing for NASA and these other places?

Herrod: This is a long time ago. He actually did his PhD during the Vietnam War, and part of the funding and everything was to help with rocket launches, and really getting involved with that. So, pretty old-school physics work on that side. He ended up then going to work for Texas Instruments, and I sort of became a Texas Instruments brat, <laughs> I guess, by moving to all their locations. What's particularly relevant today is that in the '80s, he took over artificial intelligence for Texas Instruments. At the time, it was this hot new thing called 'expert systems,' which was one form of AI. I think what's so funny is that I read articles he was working on where they claim the end of white-collar jobs, and how it was going to be all sorts of challenges, and this is 1982. So I think I've been tracking the highs and lows and winters and summers of AI for many decades now.

Spicer: Just to pick up on expert systems a bit, did he work with Ed Feigenbaum at all?

Herrod: Not directly, but Ed is--

Spicer: Was influenced?

Herrod: ...obviously a legend. Yes, certainly. He was trying to commercialize a lot of the research that was coming out. But that's where I personally heard about Ed Feigenbaum and John McCarthy, and all the legends of AI, which obviously to this day are proving to be prescient.

Spicer: Yes, and I wonder, so your dad was doing-- I don't want to call it sort of straight up physics, because I'm sure it was very complicated. But the rocket scientist, literally.

Herrod: Not too much. His was mostly, I guess you'd call it Newtonian. <laughs> That sounds really old when you say it that way. No, I mean, he got started with physics and he just became really into software management, and engineering leadership. All of our careers evolved so many times over the course of time. But yeah, he started that way and then ended up running a bunch of different divisions at Texas Instruments over the years.

Spicer: Wow, that's amazing. Now, tell us about your mom.

Herrod: Yeah, my mom was amazing. She passed away a few years ago, but she was just a total saint. She ran all these family counseling programs. I think she tried all the programs out on my brother and myself, which I got to be very attuned to: "Am I being tested here? What are you looking for?" But no, it was really great. She just taught me a real love of people and how to really get along with everyone, and how to have an open mind to different things. So, yeah, it was a really neat juxtaposition of deep science and deep empathy, and really looking at people and how to work with them.

Spicer: Yes. Would you say your dad had some of these skills as well with people? He was a manager, right? So, he must have maybe--

Herrod: Obviously some. I think they were pretty-- opposites attract a little bit. <laughs> So I think the mix was very good. But yeah, obviously he had certainly some deep empathy as well, and has. He's still alive and doing well. It was a really neat area to grow up. I have a brother who's two and a half years older, and he ended up being a straight up serious salesman, doing a lot of big sales jobs, and I was the introverted, more nerdy kid, and just thought Computer History Museum things would be my life. I tinkered with things, I loved to find out how things worked. I wrote little software programs on my original TI-99/4A computer. So, yeah, I'd say it was a family of a lot of different extremes, from sales to deep technology, from empathy to deep physics work.

Spicer: Right. So, you had the one brother and no sisters?

Herrod: Yeah, brother lives in Dallas now, and he's doing very well.

Spicer: Oh, that's great. So, in high school and middle school, to some extent, which is, I guess, when people's hobbies and interests start to coalesce a little bit, did you have any-- I mean, you're clearly interested and influenced by your parents. But did you have any really interesting hobbies, or sports, or activities that you followed, that you pursued in school?

Herrod: Yeah, I always tried to keep up with my brother in sports, he was always more athletic than I am. But definitely, I played basketball and soccer, and got really into that. I was in the land of high school football, but I was not nearly large enough to play. So, I was always doing that, but I was always more of the really very much introverted [type], into science in so many ways. It's funny, like so many kids maybe

my age, the first computer that they got had a major impact on their lives. Again, this was this new TI computer, which we had in the bedroom, and it was amazing. I could actually code little programs and I could take it apart a little bit and use the modem, making those funny noises. So that was a really major impact on my life, just realizing what you could do with computers.

That led through computer science classes early on, learning Pascal and doing all these programs, and doing quiz bowl and Mathletics. <laughs> So it was really one of those classic kinds of really scholarly upbringings, a really good school district that gave me a lot of opportunities. But on the flip side, I always had very broad interests too. I went to the University of Texas at Austin right out of high school and got a Bachelor of Arts. I always did deep computer science, but I loved chemistry, and I loved art history. I took the history of the Middle East, and I just really love learning of all sorts. I always said that computer science is like an underlying thing, but I think the world's so interesting, and getting to be part of it is really fun.

Spicer: Oh, that's really interesting that you have this Renaissance man quality with engineering and the arts. One of the famous quotes by Steve Jobs was that "It's the combination of the humanities and technology that makes our hearts sing."

Herrod: Maybe more so now than ever. I mean, as we're seeing it play out with AI and other areas.

Spicer: It's getting very personal, isn't it?

Herrod: It is, and I think philosophy and English matters as much as computer science-and math.

Spicer: Yeah, it's raising all these questions again. Great point.

Herrod: I went to the single largest high school in the United States. So, my graduating class was almost 1500 students, and--

Spicer: What name is this?

Herrod: It's called Plano Senior High. It's actually now three high schools, but this is the heart of Dallas, and the goal in life is to win football games. <laughs> So this school was custom created to get feeders from everywhere and have a giant class of great football players. But the side effect is that with a school that size, well-resourced, you can have amazing classes and all these labs and AP tests and everything.

So anyway, I went to the largest high school graduating class at that time, and then I went on to University of Texas, which I think at the time was the largest undergrad institute. I lived in a place called Jester Dorm -- which is incidentally where Michael Dell started his business -- but it was also the largest dorm in the US, with its own zip code. <laughs> So, I think I always grew up in these worlds of being a small fish in this giant pool, and it gives you opportunity for a lot of resources, to meet a lot of amazing people. I think it also really pushes you to learn to self-advocate. I think if you're not going to go out and

find something that you want to do, it's harder for someone to spoon-feed you when there's that many mouths to feed along the way.

Spicer: Tell us a bit about your later academic career, your three major steps, I think, your three degrees?

Herrod: Sure. Yeah, like I said, at University of Texas, again, I really had a great time. My whole family had gone there, and it was from a school spirit and athletics, and playing intramural sports, everything was great, and I think a really broad curriculum. I was in a really neat program there called the Dean Scholar Program, which was relatively new at the time. This was a way within a giant organization like UT Austin to get some real special seminars and special field trips. Yeah, it was 20 or so kids that were all super interested in sciences, and I have really fond memories of the opportunities from that, including going out to West Texas and the astronomy observatories. Yeah, just some great lectures and some great classes in the humanities, which I thought was a really important part of the whole program.

But I always knew that I wanted to go to grad school, and as a computer scientist, I was always looking at where the textbooks were being written. So, I had Hennessy and Patterson's book, and I had John McCarthy's book, and all these different books were something I saw every day, and I dreamed of going to Stanford without having ever been west of Texas, <laughs> really, in my life. But I was pretty focused on that. I did undergraduate research, which is a key way to get into graduate school, with a great professor named J.C. Brown, who was in computer architecture, and came, applied to and got into the PhD program at Stanford. I remember it super well. It was 1992, and I packed up my little black Nissan Sentra, and everything I owned fit in the backseat, and I drove out to California. It's a long drive from Dallas, it turns out.

But it was everything I could hope for in terms of getting through a really fun undergrad program, meeting a lot of really great people, learning a lot, but also preparing me very well for the next stage of life, which became the Stanford and sort of Bay Area stage of life.

Spicer: Yeah, that's fun. So, tell us about Stanford --you started with a master's there?

Herrod: Yeah, at Stanford, you actually get-- in computer science, you get admitted directly into the PhD program. It's not usually a progression, but you do a master's along the way, as you probably know. But I was always on the border of computer science and double E [electrical engineering]. I always liked touching hardware and building things, and that came up in my career as well. But I also loved software.

So I worked in a couple of things that were right on the border. There was this really cool project at the time called Flash, which I think might even have an exhibit here. But it was really in a time when people were doing these highly parallel computing systems, and it was if I could have one system with a bunch of CPUs tightly tied together, and shared memory, and all these different things, you could solve really high-end problems. To some extent, that doesn't exist today. It's become multi-core chips and things like that. The world's sort of moved more distributed at this point, with these giant clusters and software that manages them. But anyway, at the time, it was very interesting.

Right across the street, I collaborated with Silicon Graphics when it existed here [in CHM's current main building on Shoreline Boulevard] to build their giant machines that they were working on. I ultimately worked for MIPS, which was the CPU used at the time. So anyway, it was a great time to get involved, right at the hardware-software interface, and to meet just incredible students. Both incredible thinkers as well as founders of people who would go on to start many companies right around here.

Spicer Yeah, in a previous interview, you mentioned at one point you were sharing office space with the founders of Google, and Cisco, and Arista Networks, and...

Herrod: Yeah, it was pretty amazing. This third floor of Gates Hall at the time was where system software kind of went on. I think our Friday evening TGIF parties would have people who'd turn out to be running a lot of great companies. At the time, we were all just fun students looking for free food and beer. <laughs> But it was such a pleasure to be able to be in that environment.

Spicer: Yeah. I mean, it's really the one time in your life when you have time to think. I guess maybe when you retire, too.

Herrod: <laughter> Maybe so. It was really interesting. That leads into a lot of the other areas of life, and I think all of life is a journey where something sets you up for the next thing. But I ultimately met my advisor a couple of years in, who is Mendel Rosenblum, who's one of the smartest people I've ever worked with. He went on to obviously found VMware and do a lot of great software products. But just meeting him and then meeting all the students that he had attracted to his lab was the big changing point in my academic and ultimately my industrial career.

Spicer: Well, let's talk a little bit about virtualization as a concept, and how that fits in with your dissertation. Because that's the work product of your PhD, right?

Herrod: Yeah.

Spicer: Are they related?

Herrod: They're certainly all related. The way it worked is, so Mendel had a lab, meaning it was eight or nine students, all of us, kind of like this hardware-software interface. I'm still friends with almost all of them. I actually had a fantasy football league with, I think, all of them to this date. But we became very close working on different areas of interest, and Mendel, in particular, had a theory and was thinking about virtualization.

Zooming out a little bit, one other cool thing about graduate school, or having that much time in school is it's a unique point in time where you get to study history and look at what's happened in the past, but you also have sort of unstructured time to postulate and build things for the future. I think this is one of the best examples of it. I was actually in a seminar class run by Mendel and a few of his grad students, where you would read papers, really important papers, from all over time. One of them was about the original virtualization systems, which all started at IBM in the '60s for the most part. So, it was a pretty obvious

and interesting idea, what if you could carve up the computer into lots of little computers? Would that give you flexibility and power and all sorts of good things? So, I think the way this started is that, again, Mendel had this theory that virtualization might be ready for a comeback.

Spicer: On microprocessor-based platforms?

Herrod: But in a modern world-- I shouldn't say modern, but in the next generation of chips, which was Intel-related chips at the time. All of these were IBM custom chips, they were meant to be virtualized. So, his theory was "Boy, we're in a world now where we have Linux coming along. We have Microsoft Windows. We have these Intel chips that are becoming more powerful by the day. Could we apply this interesting problem, or this interesting solution from the past to today's problems?" This was without knowing if it could be done for sure, and you'd have to put yourself in the time. At the time, it wasn't clear that maybe MIPS or the Motorola chips or something would win. It wasn't obvious that the x86 architecture would win. So it was a bit of a speculative look. But I thought it was so cool. I don't remember who coined it. Maybe someone can bring that out. But we decided to call the project Disco, because we thought it was this cool thing from the '60s and '70s that we could bring back. <laughs>

So I think one of the early papers was called Disco, and that was one form of the solution. But there were probably five or six graduate students, each who had some angle around virtualization and how it might be applied to research problems. Mine was particularly focused on a way of, could you simulate an entire computer system and software in a way that would allow it to be studied at a very high speed, and really understand everything going on? So, that was one angle that was inspired there. There was one about operating systems. There were just a lot of different angles that people were looking at.

Spicer: Right, yeah. On this, I mean, I think the first VM/370 came out in '72, which I think was-- I mean, there were precursors to that as well, right? You could even include time-sharing, maybe, if you want--

Herrod: Yeah, I mean, you could...

Spicer: ... sort of ...

Herrod: ...definitely go back. There was a whole suite of systems that came out, and VM has always been in the names of these. So, it's not like it was a--novel phrase.

Spicer: You mentioned that IBM hardware had hooks for virtualization. Was there a cycle in which Intel had to come around and put those hooks in their microprocessors to accommodate virtualization?

Herrod: Yeah, and maybe I'll shift a little bit into the-- we can go into VMware and that creation, because I think that was a big part of it. I would say, so certainly one of the advantages IBM had, and the reason they've been such prolific inventors, they owned everything. They owned the operating system, the hardware, the applications like CICS [Customer Information Control System] and all these things back in the day. So you could take a collaborative approach with all the other divisions to creating what you needed to do.

So they had software and hardware working in perfect symbiosis to do what they needed it to do. At the time when the VMware company was started, we were a small little gnat <laughs> trying to make a big splash. What's really interesting is the whole notion of virtualization was a few things. It started as "Could I run a bunch of different operating systems on one machine?"

This ended up being extremely useful for testing and software developers. You have one box, and I can try out my software on a bunch of different versions of things. If I'm a QA department, I used to have a rack of 20-something machines. I'd have to install everything in all of them and I'd have to work on them. So virtualization carved up one pretty powerful server into a bunch of mini servers or desktops, allowed each one of them to have its own software stack, and so just a much more convenient way to consolidate a lot of what previously were a large number of machines into one.

I bring that up, part of it is the evolution of figuring out what people would want from this product. But the more important part was, as you asked, our relationship with Intel or AMD, or anyone at the time. The proposition coming in was "Hey, we have this cool product that might take 20 machines and collapse it down to 1. Would you like to help us?" <laughs> As you could imagine, we thought that was the best pitch we've ever made on earth. It wouldn't sound like the best way to get them involved. However, and this was something that was really key to the relationship and its reality, what you ultimately found is if you're going to put a lot of eggs in one basket, you're going to spend a lot more on that basket.

So, ultimately what it turned into showing to Intel and AMD was, first of all, this is good for your customers. It's going to drive demand and new use cases. But secondly, instead of selling a bunch of cheap processors, what if you sold a really good one here? You maxed it out on memory, which is a very high margin part. You only got to show everything you could do in the processor, "Would you be interested?" That actually is what ultimately got them to be really excited. It was a killer use case. It actually sold higher-end parts, and it became something they needed to really invest in. So, a long-winded way of answering your question. It took many years and many convincing things along the way. But they ultimately then, both companies added specific extensions for virtualization. It was called VT, Virtualization Technology. If we would've had that when we started VMware early on, it would've made life a lot easier. <laughs> Everything would've been faster and easier.

Spicer: But that's almost your whole-- it would've been too easy, though.

Herrod: Maybe others would have done it, if so.

Spicer: Someone else would've done it. Yeah.

Herrod: <laughs> Maybe so.

Spicer: So did you come up with software? While you were waiting for these hardware chip vendors to get with the program, you came up with software, then, to solve those bottlenecks. Is that right?

Herrod: Yeah, during the Stanford days, one of the other projects was something we called binary translation, which is something that is actually relevant in a lot of areas today. One of my lab mates spent the most time on that, and this is a way of taking code and on the fly rewriting it into other code that would have some different behavior. Today, this is how some of the original Apple software can run on these fancy new M1, M2, M3 chips. It's been used in a lot of different technologies to let something run somewhere else. So, this emulation technology was very key, and--

Spicer: Is this from your Transmeta experience?

Herrod: Well, it's interesting, and I should be super clear.

Spicer: I'm mixing up--

Herrod: I finished my graduate school.

Spicer: I'm mixing up the timeline.

Herrod: No, no, it's great. But--

Spicer: I'm sorry.

Herrod: -I want to bring it up because it's really important. I was not there at day zero of VMware. I was involved at Stanford, worked on the graduate program. I then graduated before VMware started and needed a job desperately. I just got married. <laughs> So I went to a company called Transmeta, which is also really interesting, and they happened to use binary translation as well, and they were trying to take on Intel by building mobile processors.

We can talk about that in a minute if you like, but it was a fantastic, fun time. Linus Torvalds worked there, which was just legendary. But it also let us work quite a bit on binary translation. After four years there, VMware had then started, they had done a lot of great early work, and I joined at that point. I run 50 employees in to sort of run a big chunk of engineering and help move it forward. But again, a long-winded way of saying that this binary translation was very key in the early days in particular, to how can you pretend you have a dedicated processor to yourself when you're actually sharing it with other virtual machines.

Spicer: Okay, that's very interesting. So it took the work of several grad students to be combined to come up with this VMware?

Herrod: Yeah, and I would say also <laughs> none of it directly translated. All of the work at Stanford was being done on MIPS processors, and actually we were using this-- I'm dating myself really now, but it was called IRIX, which was Silicon Graphics Unix-like operating system. So almost all the prototyping work was being done with a different chip architecture and different operating systems. Again, it was really Mendel and his wife, Diane Greene, the most amazing couple... it was the two of them deciding "Hey, I

think there might be something in applying this technology to the more modern world of industry standard architectures. Let's go for it." So they started it, the founding engineers from this lab joined, and off to the races.

Spicer: Is there a reason, and this is just a weird historical counterfactual, that this didn't spring up on minicomputers?

Herrod: It's a good question. I do think the unique point in time when this started, timing is kind of everything. It was a time when, again, the Intel processor was becoming more performant and its price performance was a total breakthrough, and it was a time when I think Linux had just arrived as more of an industry-capable operating system, that people would build real things on top of. I think Microsoft at the time, this might have been right at Windows 95, was all of a sudden having a very commercially viable operating system as well. They had also started Windows NT, which was a server-oriented one. So, I think it was the confluence of price-performance on the processors and capability, coupled with just the rise of these two, that made this look like a very forward-leaning startup, as opposed to "Let's take something that's kind of out there," a DEC workstation or something, "And apply it there."

Spicer: Yeah. Do you need a lot of memory to do virtualization?

Herrod: It definitely helps. There are some really fun innovations that, again, a couple of great engineers created that allowed-- if you have 10 virtual machines all running Windows, let's call it Windows NT 4, it turns out the memory looks very similar on a lot of them. So we did a lot of tricks to consolidate memory down. So we saved a lot of memory. You didn't need 10 times the memory to have 10 virtual machines. That being said, we use memory ourselves, and certainly when you're running a lot of different things, it helps. This gets back to the really good basket, or at least why-- not just Intel, but all the computer manufacturers were very excited to partner with VMware. Memory, it turns out, is a very high-margin part of a system, and we helped them sell more memory with all their servers. So that became a very nice relationship, commercially, because of that.

Spicer: Can you tell us how the product —use cases changed over that time, or the type of user, did that change? I guess the data center is your main target?

Herrod: Yeah, what's been amazing about this, and again, all credit to Mendel and Diane in particular, there was a vision statement very early on in the company that said, "VMware will run on every server and every data center, and it will bring these benefits." At the time, we had nothing. <laughs> So it's kind of a bold statement to say that, but it was just seeing through what all the potential benefits could be.

No startup on earth can ever go straight into production workloads and to really the highest-end systems. So we really knew that we had to start somewhere, and the company started with a desktop product. It was called VMware Workstation, and the idea there was, is there something lower risk to offer up that was very beneficial? This really did go to test and development teams for the most part. As I said, instead of having three systems, I now have one. It's super easy to switch between systems and try out my software and build it. It turns out that test and development is always a much easier to accept function

than running something in a data center. So the team that was there at the time, I wasn't there when they launched it, created this-- it was boxed software at the time, <laughs> remember that?

Spicer: Yeah.

Herrod: Selling it at Fry's [local Silicon Valley electronics store', and people would go out and use it, and it created at the time just an amazing fan base. People would come talk to you, and they're like "You've saved so much time, and this is amazing."

Spicer: Were some of them Mac users running Windows?

Herrod: At the time it wasn't Mac. Mac didn't come until many years later. This was all PCs running. It might be five different languages of Windows running on one box, because that was helpful. It might be three versions of Windows. It might be Linux and Windows. So I could use my cool Windows software while being a cool guy using Linux. All these use cases, but the idea that you could do this seemed to many people like magic. It was really the word you would hear over and over.

So that's how the company got started, and to get to your question, always the whole time we knew the goal was to get to the data center and really be running things. You have to go carefully. It started there, then we ran on print servers and file servers, things that were a little lower. They were considered a little less mission critical, and then slowly but surely we got on to higher end systems, running databases and running the core parts of a company. So that was a long march. I mean, that's five, six, seven, eight years before you're really trusted, at that point. In parallel, we were broadening quite a few things. It all started with chip virtualization and server virtualization. But I think all along we realized that there'd be a lot of benefits if you could also virtualize storage and networking, and all the parts of the data center. So that was a parallel thread going on the whole time.

Spicer Yeah, that's amazing. Was there one big sale that really put the company on the map, to the federal government, maybe, or a giant institutional customer?

Herrod: Oh, it's funny you mention that. I actually remember it very well. I was walking around my neighborhood and a salesperson had the first million-dollar sale that we ever had. I'm pretty sure this is all correct, that it was a million-dollar sale, I believe to Merrill Lynch at the time. I remember a big celebration and us thinking "Wow, we sold a million dollars to a financial firm," which is usually very discerning. I may be a little off on one of the details there, but it was something roughly like that, where we all really just were amazed. As a startup in Silicon Valley too, the first time you see yourself covered in a newspaper, that's amazing. For me, I remember in my CTO role there, I would do a lot of travel around the world talking to user groups, or potential customers. I remember arriving in Eastern Europe once and seeing an ad for VMware, and it just blew my mind. I couldn't imagine that this--

Spicer: Nice feeling.

Herrod: -thing that we worked on was in Eastern Europe, and people loved it.

Spicer: Since that was such a big part of your job, since you bring it up, the travel, which for most people is really exhausting, depending on the tempo of your job. Did you find it tiring? I mean, you're a people person, so I know you probably enjoyed it to some extent, but...

Herrod: I think anyone who's traveled a lot sees it as a grind at times. But I mean, honestly, I couldn't think of a better job on earth. I think I became CTO kind of halfway into my career there, and some people joke that it's called chief traveling officer, <laughs> and others say it's chief talking officer. I'm not sure, maybe it's both. It was exciting. What was great is people cared a lot about VMware. Ultimately, we had offices all over the world with employees, and I was running engineering at the time. So a lot of engineering, recruiting, and helping the employees know what's going on. We had partners of every sort, technology partners, and channel partners. So it was convincing them to stay with us and telling them the future. We had big customers that we were going to. So it was just getting to spread the gospel of VMware and what we were doing. It was a really privileged position, but it was a lot of travel, <laughs> that's for sure.

Spicer: In another interview, you said one of your bosses told you that as you move up the corporate ladder, the need to communicate increases tenfold every time-- every step up the ladder, you have to say it that many more times to that many more people.

Herrod: I found that to be completely correct. <laughs> I do think as you get to these big organizations, it is all about communication, and being crisp and doing it over, and finding ways to communicate in a way that best resonates with them. Some people are different types of learners. But I've been fortunate to always be able to connect with people fairly well. Again, I think communication is kind of everything in this role, in particular. So that was a very standard part of it, and a lot of public speaking in terms of VMware created this amazing user conference, which just finished in Barcelona actually, last week. It was called VMworld when it launched, now it's called VMware Explore. You'd get 20,000, 30,000 plus people coming to hear about your roadmap, <laughs> which is kind of stunning when you think about it.

Spicer: That's exciting, yeah. With VMware, what was the competitive landscape like?

Herrod: Early on, there was no competition, which was pretty fun and exciting. It has a different set of problems because you're having to convince the world that this can actually work and that it exists.

Spicer: It's a new category, isn't it?

Herrod: It's absolutely a new category, and it doesn't have a name, it doesn't have a budget. People are pretty leery whether it works. So there's a set of challenges, but one of them isn't a competitor. <laughs> So, that makes it sort of a different set of hard. But the big thing that happened, Microsoft bought a small company. I think it was called SoftPC? I'll have to double-check. I've forgotten all the way back when it was, but this was 25 years ago, 22 years ago, and it was both a great thing and a scary thing for VMware. When you hear of Microsoft coming into your space, especially at the time, it was one of the scarier things you could hear about, with all of their resources and their reputation, and being able to bundle it with other software.

So that was a really big moment for us. At the time that happened, we were getting a lot of companies calling us to say “Could we buy you? Could we partner deeply?” Because we were making some traction in the enterprise at that point, and that definitely had an influence on us choosing to be acquired by EMC at the time, which was already a very strong enterprise software brand who understood what virtualization could become. They had worldwide offices and salespeople, and all these things that could help. So I think we definitely had the scare of Microsoft coming into it, and there were other good reasons to be acquired, but part of it was realizing “This is serious competition now. We need some serious partnerships and big brothers and families to be part of.”

Spicer: There are many reasons for the link up with EMC, but was virtualizing storage part of the transaction?

Herrod: It definitely ends up it is. So, I'll backtrack for a moment. A really, really important part of the VMware value, and value proposition that we'd sell to everyone, is that we are agnostic to the underlying hardware. If you want to be a Dell customer, or an HP customer, or an EMC, or a NetApp, we're going to virtualize all of it. We're going to make all of them work great, and we don't want to force you into choosing something. We want to make sure you have choice. I think around that time, virtual machines, as I said, they take up a fair amount of memory. They take up a lot of disk space. So we were driving a lot of really big sales of shared storage for the first time, in which, at the time, EMC was a major player, as was NetApp and others.

So I think they naturally saw us, and they saw us as a great accretive thing. They also saw us in their accounts as something growing. We had a lot of options for being acquired from many different companies, but EMC had a very strong proposition on leaving VMware independent, and making sure that we could be a Switzerland, supporting all of the different hardware vendors at equivalent levels. They stood by that and that was a really important part of it, and that was part of why we thought that would be a really good match as well.

Spicer: As far as customer support, and we're just bouncing around a little bit here, I assume there's like a VM Academy or something along those lines to bring people up to speed, on how to use the system, and...?

Herrod: Yeah, and that's a really good point to bring up, and I think this is lost a lot of times in companies. So, there's certainly customer support, and all the things we did to make sure people were successful. But usually when you bought VMware, especially in the enterprise, in a data center, you wanted people trained in how to use this properly. What's more, people, individual employees, we wanted them to feel ready to do this right, and it would help us ultimately have a higher rating, and have better customer support.

So we did create a pretty nice certification early on, on server virtualization, and you see that today with a lot of other fields. It ended up being so many companies were getting interested in this that having that on your resume was a hugely important thing to do. So you'd find one company going to pluck the server

virtualization guy out of another company, and so that's a really nice sort of feeding both on wanting people, or people who want to be trained. They saw it as an advancement to their career.

But it also means someone who's an expert in VMware at one company goes somewhere else, they're likely to buy more VMware when they get there. So the training programs I think we're very well regarded. We had some great people working on that space even to this day, and it ended up being I think just a win all around. It helped the employees who took it, it helped VMware sell more products, and it helped the products do better once they were purchased.

Spicer: Yeah, absolutely, and the true support happens after the sale, right? So, one of the things you mentioned was you called VMware a Swiss army knife for-- this is in a previous interview, I'm sorry, that I'm referring to. Can you explain what you meant by that?

Herrod: I should say, first of all, one of the founders of VMware is a great guy named Ed Bugnion, who was from Switzerland. <laughs> So he liked to talk about Swiss things all along. But I think it actually means two different things. A Swiss army knife, as everyone knows, or maybe-- I don't know if kids still use them, but they have all these gadgets and they can do a lot of things. We found virtualization itself would be adopted for one reason, maybe for test and development, but it turned out it was super useful for server consolidation in the data center, or it actually opened up the door for a great disaster recovery plan. There were all these different IT functions that were made better by using this core underlying technology. So in that sense, it was all the gadgets on a Swiss army knife.

The flip side is, I guess, more about Switzerland itself, which was we were very, very careful to make sure that all hardware ran great when running VMware software, and we had to create all sorts of certification programs and really invest heavily in that. That was really, really important. The notion that people would adopt VMware software and it would ultimately give them more choices in what they would buy hardware-wise was really empowering, and it was a very big part of our value proposition as well. It would give them some more options to buy the right hardware at the right time, at the right price, and that ended up being very strategic. So I'll call that Swiss neutrality, <laughs> and the other part is Swiss army knife.

Spicer: Okay, that's good. Yeah, because you can virtualize different things, right? It's storage and memory and all that. So, those are some of the little functions on the knife, I guess.

Herrod: Yeah, that was the scissors or the toothpick <laughs>.

Spicer: Exactly. Tell us about the modern data center today, and how VMware is used. Is it used by the usual suspects like Meta, and Google, and that scale of company, or...

Herrod: There's a few different pieces to it. The core virtualization itself, so as this was all growing up, that is when cloud computing itself started to come alive. I remember a very early conversation with Amazon as they were launching this crazy new AWS thing, and at the time, our goal was "Hey, you should use VMware software." But as I think it would play out, most of the really high-end cloud services,

whether it be obviously a Google or an Amazon, or even Microsoft Azure today, they're not going to likely pay some outside vendor a lot of money for something running literally on millions and millions of servers.

So, along this time, there were open-source efforts ramping up around virtualization, and there's one called Xen that you might've heard about, and there were others over time. There are also other types of virtualization coming along with things that are today called Kubernetes, or different forms of that. So, for the most part, the highest end cloud services coming along were using virtualization, but not from VMware. They were taking an open-source version and heavily customizing it to their own unique needs and challenges. So that was going on for the highest end of the cloud services. But big telcos and other companies running these big clouds of their own would definitely adopt and use our software very heavily.

To this day, a lot of corporations, they certainly have some things out in these public clouds that they're running. But the crown jewels and so many things that they run every day and rely on for businesses are not running in those kind of public scale clouds. They're running on traditional servers, running VMware, and that sort of software. I don't know what the split is today, but it's a very heavy-- you'd call it in their on-premises data center, even though they're running different places, versus running in some public cloud.

So, there's a lot of work around, and especially where VMware's roadmap has gone long after I left, is really thinking about in a world where you have these multi-clouds everywhere, what value can you play, and what can you do in that world? I think if you just think about it as a very logical progression of I virtualize a few servers and let them really work well together, maybe I could virtualize a few clouds and have them work really well together. Different abstraction, but the same core goal.

Spicer: How does cybersecurity fit into virtualization? How do they talk to each other?

Herrod: A really important question, all the way from when we started, to this very day. Security, as you know, is <laughs> the core of all challenges in almost all data centers. For the most part, VMware has never really had a core this-is-a-cybersecurity-product. But people would adopt the data center product in particular for security reasons, and I'll give you a couple of examples.

First of all, when you're running all these servers and you can have them moving around between machines, it gives you an opportunity to do things you couldn't do before. You can actually very easily just turn them off when they're not running and quickly spin them back up, and that's really important. It's hard to attack a server that's off. <laughs> So that was actually one thing that we did quite carefully. The notion of a virtual machine also made it much, much easier to update software, and that is also one of the main ways that different computers are broken into today. They have an out-of-date patch system. I'm jumping into the tech a little bit, but because a virtual machine, it used to be a piece of hardware, you'd have to turn it off, you'd have to install these updates, you'd have to test it and bring it back on.

In a VMware world, you could actually take a copy of that server, literally copy a file from here to here, apply all the patches to this new virtual machine that's running over here, test it all over here, and all the time, the old one is still running in its own environment. When you're ready, you simply plunk this thing over the old one and now here's an up-to-date system with all the security patches in it. So that's a little

bit arcane, but it is another example of why things became safer when you're doing this, and especially as we got more into network virtualization, and the ability to do interesting things with firewalls, and a lot of other data center technologies, that also made security substantially better in a VMware world.

Spicer: I'm curious about the way that jobs are scheduled, and is it that, like a round-robin thing? Everyone gets a time slice? You can see <laughter> this is a very historical point of view I'm taking. Or is it spread out over a...?

Herrod: It's a great question. The answer is yes. There were two different things just to sort of set the stage. One is when you're on a single server, you have, call it 15 virtual machines, each of which thinks it has its own server to itself. So, just like a lot of operating system abstractions and a lot of things you do in the cloud today, we had very easy ways of saying which ones were the priorities, and actually it got pretty nuanced. We could say which one had the priority for memory versus for CPUs. We could do guaranteed minimums versus just sort of timesharing.

So I'd say it was a very rich set of rich set of abstractions for that. But what was really cool, in the early days, it was one server running a bunch of virtual machines. What we adapted to pretty quickly was "Let's take a bunch of physical servers and have a bunch of virtual machines running on top of all of those." I'd say one of the biggest magic things that we also created was something we called vMotion. That meant that you could take a live running virtual machine running on physical server A and move it to physical server B without anyone from the outside ever knowing it. So it was live, and we did all these great demos of actual database transactions still running as you move this live virtual machine from one place to the other. That opened up the next door on resource management and on scheduling.

Spicer: And like load balancing...

Herrod: Absolutely, load balancing. Here's a really cool thing that we could do. Because you now abstracted all these physical servers from all these virtual machines running on it, at nighttime, if things got less busy, we could actually consolidate them down to one server and power off everything else. As the morning came up and things got busy, we'd power up some physical ones, spread it back out, and that ended up saving a substantial amount of power.

Overall, I think one thing not covered heavily in VMware's history is how much power savings came as a result of what we did. So anyway, it's a long-winded answer on the resource management, but it was the ultimate flexibility in terms of where you could do things. We were even able to move things across data centers while still running, and the promise was always that we could get something closer to where the users were at a later time and day, where we could actually go to a place where maybe power was cheaper at that time of night. This flexibility is something no one ever had in the early days of servers that were running physically.

Spicer: I'm sure they're concerned with that now, with data centers, with by-the-minute operating costs.

Herrod: I mean, power to this day becomes a major part of it. This idea that you could just move your running system to the best place for it, it turns into power savings, cost savings, best latency for users. A whole slew of things if you aren't constrained by physical boundaries that you might do.

Spicer: Right. Oh, that's fascinating. I'm going to flip back to the engineering, your CTO job, chief traveling officer.

Herrod: I was in the CTO role for a large chunk of time, and I was also in the VP of Engineering role, and they're not necessarily the same thing. But I think at the very peak, I had that many people rolling up into my organization, which is a whole other discussion topic, perhaps, on trying to manage that many super talented people and go through it. But as always, it's done with a lot of managers that you have that are carrying the load and doing a lot of work. But it was just trying to think of that many people with different career aspirations who need to understand what needs to be done, who have good days and bad days. From a people perspective, it was really exciting and interesting, and really difficult.

Spicer: Remember what they used to say about never buying a car built on a Monday because the assembly line workers were still hungover?

Herrod: They had a tough weekend? Yeah.

Spicer: Yeah.

Herrod: <laughs> We had a few rules. I don't know if this is the closest analogy, but we actually had a rule where you couldn't check in software on happy hour night. It was kind of similar. <laughter> So, we had a few rules--

Spicer: That's a good rule.

Herrod: -and I guarantee there's always a story behind a new rule, so...

Spicer: Oh yes, it's like every law, behind every law, yeah--

Herrod: Something happened.

Spicer: -is a story <laughs>. Yeah, exactly.

Herrod: Yeah. It was a great team though. I mean, this is spread across-- I'm not sure at this point, but at the time, I think we had 16 or 17 engineering offices around the world. The sun never set on people doing engineering and testing. It was just a humbling experience to be surrounded by that many amazing people whose salary and their livelihood comes from this company. It was really neat.

Spicer: Wow, yes. How would you describe the first 10 years in terms of growth? Pretty exponential, or...?

Herrod: Pretty exponential. First, it was all about hiring great people, and you'd hire someone great, and the next person, they'd bring their people. So there's a little bit of exponential of you hire a great person, and they come with connections. Relatively early on, we moved into acquisitions as well. We had a pretty big vision for what we could do, and by the time I left, we had made about 25 acquisitions of all sorts and of all sizes, some of which worked, some of which did not. But it was definitely a big part of our growth as well.

Spicer: Oh, yeah. I wanted to talk about acquisitions.. What would you say were some factors that you used to evaluate potential acquisition targets?

Herrod: <laughs> Do you want to hear when we did it well, or when we did it poorly?

Spicer: Well, both, I guess. They're both instructive, right?

Herrod: They are both instructive. Early on in a company like this, you're at a company and you meet some technologists and they're doing something, and it's interesting. You're like "Boy, if I had that here, I could sell it and it would be amazing." So, you spend a lot of time thinking about the technology and what's accretive and how it would work. I think, especially on the engineering side, what you find pretty quickly is that the cultures and how they mesh, how your different systems are going to align.

If this is a company shipping stuff every week and our server products need to ship every two years, it's something you have to test really carefully, you can get some real things that are out of sync with one another. So, I do think, me personally, I made plenty of mistakes, but in the early days, technology would be like "We've got to have that." Later on, you really learn that you have to think "How does this fit in with our existing products?" Very quickly you learn "Is this something that our sales team can sell? Are they even talking to the right people?" As you get very large as a company, you realize "Hey, even if this works out great, it's not really going to move the needle on what we need to do. Is it more of a distraction than a benefit?" So, I would just say, I certainly learned all these lessons by watching which ones were great and which ones were less good on that front.

Spicer: When you were hiring or merging with these people, were you after technology or people?

Herrod: It's always both to some extent. You wouldn't typically buy a technology if you couldn't sustain it.

Spicer: That's true. Yeah, it's tied to a person, for sure.

Herrod: But we actually did one pure people acquisition. If you're interested, we can go into that? Generally, it was a cool product in an adjacent area that we thought would really help accelerate our roadmap, and we thought the people were great, and so that was a lot of it. I would say what you shouldn't overlook, maybe in the hybrid world that we're in today, post-COVID, it's different. But we ended up with offices in Bulgaria and in Southern France, and in a lot of areas where we didn't have anything else. I think if you're not really committed to being there all the time and keeping people in the loop, that team's not going to feel really part of the overall team, and you might lose them over time. So I think I'd be

very curious if things change these days when so much is over Zoom and Google Meet and that sort of thing. But at the time, I think our eyes might have been bigger than our stomach on a few cases. <laughs> We thought “No problem, we can bring them into the fold,” and didn't realize all the extra costs involved with doing that well.

Spicer: Right. I'm thinking of Cisco, for example, as a great example of a company that is known for its acquisitions.

Herrod: I can't even imagine what the number is now.

Spicer: There's a wiki page on their acquisitions. It's--

Herrod: Hundreds I'm sure, at this point.

Spicer: Yeah. So that's interesting. Did you find it very useful in terms of any of those acquisitions? Did they really give you a big boost--

Herrod: A few of them--

Herrod: -Some of them were amazing. The very first one we did was actually really cool, very small, and it was a team that was doing a project. It was a product that you would throw into a company's-- you'd actually turn it on inside a company wall, and it would actually go out and sniff all the servers that they had, and it would write a nice little report that says “You have 10 physical servers running. Did you know they're all only running at 10% of utilization? Boy, you could use VMware. That'd be great. It would save you this much money.”

That was kind of the product they had. It was sort of a consulting business, and it was the best acquisition of all time. It was a tool that we then used every single customer to just let them know the benefits of using VMware, and it was something we could really use for lead generation. So, that was a home run. Not deep technology but just a nice product and a good set of people that joined.

The last acquisition I was involved with was a really big one. It was called Nicira, and it was in the network virtualization space, over a billion-dollar acquisition. That was a really important one for VMware. We had our own network virtualization work going on, but this was a well-funded, very strong team that knew what they were doing and was getting some early traction in the space. We ended up fighting Cisco tooth and nail to acquire this company, and it ended up being what is today a billion dollar plus revenue business at VMware, and a real core part of their data center offerings. There were 20 something in between there that I'd say <laughs> had mixed ups and downs of different sorts, but I think those were two really meaningful ones.

Spicer: Can you tell our audience, and by extension, me, <laughter> how you virtualize a network? How does that work?

Herrod: It's a similar concept. It's a piece of software that runs in front of the hardware and it has to be very high performance. It has a lot of the traits that you would have to have for this. You remember this maybe from your own schooling, but there's an old saying that you can solve all problems in computer science with another level of indirection, and that's exactly what this is. A piece of software in which in the old days a network packet would come in and be routed directly somewhere else by the hardware. Let's say you could put a little piece of software on top of it. You could have it come in and the software would quickly say "Eh, I actually don't want you going down that path. I'm going to send you somewhere else," and that's a simplified way of thinking about it. But it is a layer of software that can separate logically what you're doing on the network from physically what those underlying hardware pieces are. It was Software Defined Networking was sort of the catchphrase at the time, SDN. It was basically the promise of doing everything VMware did for servers but for the network, and VMware decided "We should do that."
<laughs> So, that was how it played out.

Spicer: I play around with ham radio and, of course, software defined radio is huge.

Herrod: It is, actually.

Spicer: Yeah. Are you a ham too, or...?

Herrod: I'm laughing because there is a great team out of Boston that was doing software defined radios. It was the Bose family, of Bose speakers. But I know even to this day, everything I look at, I'm like "If you virtualized that thing, you could solve a lot of problems."

Spicer: <laughs> Yes, exactly. Okay, I wanted to talk to you now about the VC phase of your life, and tell us how that started and what drew you into that?

Herrod: I was at VMware for about 13 years or so, I think, 12 or 13 years in this role, and I had really enjoyed it and done everything. But we kept doing acquisitions, so I kept meeting these startup teams. I guess two things, I'd see this young company and I kind of missed the earliest days of companies. I really love right when you're getting started, infinite promise and possibility. Hiring, all this stuff is so much fun. So I missed the earliest stages of companies. But I also kept meeting the <laughs> venture capitalists that were representing these startups, and they were very smart people. It seemed like they were enjoying life a fair amount, and learning quite a bit.

So, I thought after 13 years it'd be fun to take a try at that and get back to the entrepreneurial roots. The thinking, at the time at least, was that there weren't a lot of real deeply technical kind of CTO types that had gone into venture capital. So the theory was maybe if I really understood these core technologies more, or could be talking about building engineering teams, that that would be something that would help me either meet startups earlier, or convince them that we were a good partner. So that was the theory, and then I was fortunate to meet a lot of really interesting VCs before settling on General Catalyst, which, actually, ironically, Diane Greene referred me to Hemant Taneja, who runs that firm. To this day I respect every opinion she has. She said, "You should take a look at them." It ended up being an upstart just

moving out to the west coast, and I thought the idea of being at an entrepreneurial venture capitalist would be a lot of fun.

Spicer: Well, that's neat. What's the difference between an entrepreneurial venture capitalist versus a non-entrepreneurial?

Herrod: In this particular case it meant they weren't here yet. They were literally arriving on the west coast. They weren't doing early stage investing in the areas that I was interested in, data center software. So it was entrepreneurial in the sense of part of this didn't exist yet and it was a chance to build it and develop a reputation, and do our first investments around that.

Spicer: Right, and I think is it Illumina?

Herrod: Illumio is--

Spicer: Illumio. Was that your first one, and it was actually a home run, I think. Is that right?

Herrod: Yeah, I'm actually still on the board there, and it's an amazing company--

Spicer: Tell us about that.

Herrod: ...not more than a mile away from here.

Spicer: Oh, tell us about that please, and how you got into that, and...?

Herrod: Coming over into venture capital, I certainly knew the VMware roadmap, and I knew what we were good at, and I certainly knew customers. There was always this idea that there was a way to do even more around virtualizing networking. We were just talking about Nicira and that final acquisition. Right when you show up at venture capital office you were like "What do I do now?" <laughs> So I started reaching out to people that I knew and got introduced through one of the original VMware people to someone who was looking at this as an entire new project, this idea of virtualizing all the networking and doing interesting things with it. So that was a great introduction. It was a great founder named Andrew Rubin, and he to this day is the CEO, and--

Spicer: Andy Rubin?

Herrod: They're actually both named Andy Rubin. There's one who started on Android, and this one <laughs> started Illumio.

Spicer: I know the Android. Okay, yeah.

Herrod: Yeah, they live near each other as well. So, just to make it confusing.

Spicer: That's funny.

Herrod: But anyway, he's been a great entrepreneur. They have a multi-billion dollar valued company at this point, 500 engineers, or 500 employees.

Spicer: It is a software product?

Herrod: It's a full software product for the cloud, and for a modern version of the networking world, and it's another form of virtualization that's proving to be very heavily used for cybersecurity. But yes, that was my first investment, and I'm still, even after I've left General Catalyst, deeply involved with that team, and love it.

Spicer: That's really good, and is the team still the original team for the most part, or...?

Herrod: For the most part. A few changes here and there. Andrew still runs everything and they're in Santa Clara. They're just a couple miles from here.

Spicer: I had a VC tell me once "The first question we have is when can we get rid of the founder?"

Herrod: <laughter> Yeah, that might be the tough view of it. It's very rare. I mean, just categorically, it's very rare for someone who has the DNA to start something from zero to also be running a multi-billion-dollar company with salespeople, and all different things. But ironically, Diane Greene was another person like that, from day zero until she left the company. So, yeah, you don't often find someone that scales from the start to these giant companies.

Spicer: Yeah. Well, okay, tell us about Datto?

Herrod: Datto is this other--

Spicer: Datto.

Herrod: -really interesting-- that was my--

Spicer: How do you spell that, by the way?

Herrod: It's D-A-T-T-O, and it was designed to be a combination of data and ditto, like the same. All these stories you hear from venture capitalists, or I don't know, I find them deeply interesting. Other people might glaze over a little bit. <laughs> But it's all about finding an amazing person at the right time doing something amazing. In particular, a gentleman named Austin McChord is still to this day one of the most fascinating people I've met. He was not great at school. He would be the first to say that no one would hire him. So he had to start his own company. <laughs> In Norwalk, Connecticut, of all places, he started hearing about backing up data, and said "I could do this better and different."

He started this company that helped other companies just really easily back up all their data, and he did it in a really novel piece of hardware, and in a novel channel. He sold through what's called the Managed Service Provider, MSP world, and he built without any mentors and without any real knowledge of how to do startups in Norwalk, Connecticut. He's building this super-fast growing amazing company. I was fortunate enough to get introduced to him and I flew out to Connecticut, and the things that Austin did are just to this day were stunning.

I remember walking into his office, and the phone systems back when you used desk phones, it had a little screen on it, and he had the minute-by-minute sales results that he had hacked the phone to show. He had just all these metrics everywhere, and he had just found novel ways of starting this company. It was his high school friends around him. This was one of my favorite stories. He needed to build a customer support team, and so he went out and he visited the local Apple stores. He would go to the Genius Bar and offer them \$5 extra per hour to come work for him. So, I think he got banned from the Apple stores after <laughs> doing that.

So just an amazingly entrepreneurial founder who is building a great business, and myself and a gentleman named Paul Sagan, who is the former Akamai CEO, we managed to get to partner up with him and do an investment. That company sold just recently for I think over five billion dollars.

Spicer: Five billion?

Herrod: Five billion.

Spicer: Oh my gosh.

Herrod: So, just an amazing ride with an amazing founder as well. So my first two <laughs> investments were Datto and Illumio, and I was like "Wow, this is really fun." That doesn't always happen, for sure.

Spicer: Is VC your full-time activity these days?

Herrod: Yeah, so I actually left General Catalyst recently. It was a fantastic run. I spent a decade there doing early-stage investing, and I'm still very close with everyone. I'm still on boards there. But I also was looking for something a little bit different. In my time there, it was 35 investments, and after that, I was on 15 boards or so. You realize pretty quickly, it's amazing how some VCs are able to do this. But my brain is a little bit more like "Let's do fewer things and do them deeply." So anyway, that was my real goal, was to find something much more like the early days of VMware, than fast moving, lots of deals and VC. So, I left about a year ago and was on a search to find this perfect mix for this third stage of my career, and that's what landed me at a company called Juxtapose, and I've been working for about eight months there now.

Spicer: Tell us about that. It's for very early-stage companies?

Herrod: I'd never seen this model before. It is a firm that all we do is create new companies, and there's a lot of models around Silicon Valley of incubators and studios, where you're helping people come into your studio, and help them create their ideas. This is really focused on using a lot of just really smart people and doing diligence yourself on lots of ideas, ultimately starting the company yourself and getting it going, and then very rapidly bringing in just a killer team that's custom built for that idea. So I would just say it starts with a very well-built out thesis, an idea where you've done literally hundreds of customer calls, and done a lot of research. So, it's a little bit of a different order than you're used to. But we've only started 12 companies to date in 6 years. It's very focused on doing a small number of companies really, really well. So they've done a lot of healthcare companies and a lot of other areas. I'm joining to bring a software business mindset to it and start companies in the software space.

Spicer: It's a \$450 million fund, I think?

Herrod: Four-hundred-and-fifty-million-dollars under management, and that is a lot of money for starting companies, with a couple of million dollars. But the idea is that we start them, we really get going, and then we back them for a long time. So, yeah, it's different from what I had seen anywhere, and again, for me personally, I want it to be just very, very hands on, on a small number of things, but still get this really interesting idea of an investor getting to see a lot of other ideas along the way.

Spicer: Yeah, and in another interview, you mentioned three years is kind of the ideal period for a company to get its product out the door.

Herrod: It can be, especially in the enterprise space. By the time that you're creating something that satisfies all of the certifications and security needs, and all these different areas. I would say that that timeline has accelerated a lot for all companies relatively recently, for a few reasons. One, starting a company now is there's so much more that you can work with. There's cloud computing where you don't have to build a data center. There are hundreds of open-source software packages that you can just get going with. The cloud software out there helps it make it easier than ever to write things. AI now can help you <laughs> get your code going faster than ever.

So I think the speed of building things is faster than ever. Making sure you understand what the customer needs and that you work with their existing systems and that you go through their own certification trials. That's sort of a human speed, which there may be some ways around it, <laughs> but it's something that just takes a while before you're adopted in the biggest companies. Especially things in the finance space, or health care space, they're very conservative for a reason.

Spicer: Yes, and I know AI and radiology, for example, is just-- they are really close.. I think it's actually useful, rather than phrasing it as "Oh, it's going to put radiologists out of work," to say, "it'll put radiologists in places where there aren't any."

Herrod: I think that's a very healthy way of thinking about it. I generally think of AI as a superhuman strength for us as people, and in general it can take care of a lot of things that maybe we're not the best suited for. Things that are the same thing over and over again, and actually allow us to focus on the

unique cases, or pushing the envelope on new things. So, I think even for radiology, it might be-- it does a great job of looking at these slides and recognizing things you've seen from the past. Maybe it frees you up to certainly treat more patients and help more people, but also maybe to think about new things that you could find and new ways to train AI.

Spicer: The key concept behind AI, I think, is trust.

Herrod: I think we're going to hear a lot about, yeah, AI trust, and trust is on a bunch of fronts. Will it keep my data private? Will it give me an answer that's correct? Will it self-propagate and do something Terminator-like <laughs>?

Spicer: Have you heard the term "Stochastic parrot"?

Herrod: I have not. Stochastic --

Spicer: That's what some people call ChatGPT, right? It's..

Herrod: Stochastic parrot.

Spicer: It knows what you're going to say next because it's processed a gigantic corpus of the way people speak. So--

Herrod: I like the term. I hadn't heard that before.

Spicer: Yeah. <laughs> It's from a paper critiquing Large Language Models, of course, saying, "It's very good but it has no idea what it's saying." That's the thing.

Herrod: Sometimes that's true for humans as well.

Spicer: It's misleading. <laughter> Yeah, exactly. We might become parrots ourselves. All right. Oh yeah, there was one interesting tidbit you mentioned about swimming from Alcatraz. Tell us about that, and what you do when you're not ...

Herrod: Yeah--

Spicer:...master of the universe sometimes?

Herrod: It's funny, like you, I went to Stanford, and I think in 25 years since then, I've moved like a mile and a half away. <laughs> So, I still live in the Bay Area and love it here. But no, I have an incredibly rich personal life outside with friends and family. I have three kids that are college age now, and I think I love exercise to this day, which started early on. But I try to golf and swim quite regularly. Those are my outdoor things that I like to do. But yes, I've tried to do some very long-distance swims as of late, and I'm

targeting a swim across the Alcatraz Channel, which is a very Bay Area thing to do. You wear a wetsuit, and you hope there are no sharks, <laughs> and you get out there. But--

Spicer: There's someone with you, right, in a boat?

Herrod: There's always a support group looking for jellyfish and other bad things.

Spicer: Oh, jellyfish too, yeah.

Herrod: <laughs> That's right. But you have to pick the right time of year when it's warm enough.

Herrod: I think so. It's a couple of miles, and it's choppy and cold, and--

Spicer: Oh my gosh, saltwater.

Herrod: And jellyfish, exactly.

Spicer: Well, thanks very much. Was there anything else you wanted to let us know about?

Herrod: Boy, no. I mean, I'm flattered to be here and listen to this. I've watched a lot of amazing people on these videos. I'll just say I'm super fortunate to have -- probably started with fiddling with a computer when I was seven, and next thing you know, you're in Silicon Valley doing companies, and it's just a real privilege to be on this path.

Spicer: Great. Thank you very much.

Herrod: Thanks.

END OF THE INTERVIEW