Behind the Tech Kevin Scott Podcast

EP-12: Bill Coughran: Human-Centered AI

BILL COUGHRAN: What I've always done in the last many years is give you my advice as a leader inside a company, but then I will also give you my advice as an individual. And I think, to me, that's been one of the things that I think has bonded people to me over the years.

[MUSIC]

KEVIN SCOTT: Hi, everyone. Welcome to *Behind the Tech*. I'm your host, Kevin Scott, Chief Technology Officer for Microsoft.

In this podcast, we're going to get behind the tech. We'll talk with some of the people who have made our modern tech world possible and understand what motivated them to create what they did. So join me to maybe learn a little bit about the history of computing and get a few behind-the-scenes insights into what's happening today. Stick around.

[MUSIC]

KEVIN SCOTT: Hello and welcome to Behind the Tech.

CHRISTINA WARREN: I'm Christina Warren, Senior Cloud Advocate at Microsoft.

KEVIN SCOTT: And I'm Kevin Scott.

KEVIN SCOTT: Today, we're going to be talking to a person that I have long admired, Bill Coughran. I met Bill when I was a young engineer at Google, and he was one of the executives running all of the very important bits there. Ultimately, Bill went on to run Search, Infrastructure, Security, Maps, Local, so this huge swath of all of the big, complicated, important things at Google.

And like in a prior life, he'd been an entrepreneur, he helped run a networking company, and before that he had been a computer science professor and was at Bell Labs running the computing sciences research division during a period where they invented things like the C++ programming language. So, I still to this day am slightly in awe of Bill, and he is the engineering executive I would like to be when I grow up. (Laughter.)

CHRISTINA WARREN: I don't blame you.

KEVIN SCOTT: So, I'm curious, Christina. Just like everyone else that is on the show, including me, you've had not necessarily a straightforward path into technology. I was just wondering, like, you know, can you tell us a little bit about what that path has been and which mentors you've had to help guide you as you sort of figured out your own journey?

CHRISTINA WARREN: Yeah, definitely. So, before I joined Microsoft, I worked in media for a decade, and I was a journalist and an analyst and reported mostly on technology and business stuff. So, I did have tech and I've always had an interest in tech, but it was obviously a different perspective.

And what's kind of great when you ask about mentors, I think my -- probably my biggest mentor would be a man whose name is Jim Roberts

He was my editor-in-chief at Mashable for a time, and he previously was the deputy managing editor of the New York Times, and he's now currently the editor-in-chief and chief content officer at Cheddar. And just he's got like this great longstanding like old school journalism career.

But what I love about Jim is A) we used to fight all the time, so he really kind of pushed me, right? Like we used to fight all the time. But B) he saw what was happening to the media landscape, and he decided that he wasn't just going to sit back and not evolve. You know, he was early on Twitter, he was early on embracing social and digital platforms, and really pushed to get the New York Times online and to make -- to bring the two divisions that used to be separate, like to bring the print and the online part together.

And so, when I switched careers, he was somebody that I kind of looked at, because he -- even though he didn't switch careers, was willing to evolve and wasn't just going to stand back and hope that things continued the way that it'd always been.

And so, even though we used to always fight, like I learned so much from him, and I continue to learn so much from him.

KEVIN SCOTT: Yeah, the thing that really always amazes me and inspires me about people who are willing to spend their time mentoring others is that-- these people have extraordinary generosity in sort of taking the time and then sort of sharing their perspective. And like to be a good mentor, I mean, it sounds like because you guys were like fighting all the time, like that requires a lot of energy. Like that means that he cared enough about you and your success and what you were doing that he was willing to like fully engage with you and push, which is not an easy thing to do.

CHRISTINA WARREN: Not at all. And I have to say the most gratifying thing is that in recent years he's actually reached out to me for my input on different things, which has just felt amazing. And I know that I can do the same with him, so yeah.

KEVIN SCOTT: Yeah, I bet that is -- that's really awesome. And one of the things that we're going to hear in my conversation with Bill is like Bill was one of the most influential mentors in shaping my career path. Like there were many, many moments where the advice that Bill gave me was perhaps the pivotal thing in like these very pivotal decisions that had these sort of long-term repercussions for my life, and my family, and the people that I worked with.

And you know he was incredibly busy at the time, and like he was incredibly senior relative to me. He somehow or another decided to take the time, like and I will -- I will always be grateful for him.

And I've sort of had the same thing with Bill that, you know, it sounds like you have with your mentor where, I will try to return the favor to Bill and like help him out with some things that he's working on.

But like maybe the -- you know, sort of the biggest reciprocity there is that I was so moved and so appreciative by what Bill was willing to do for me that like I try to do the same thing for other folks. He's really informed like my philosophy on coaching and how I give advice to people, and like he's one of the reasons why I more often than not will say, you know, yes when someone asks me for like a little tiny slice of time for advice.

CHRISTINA WARREN: I love that. I love that so much. And I -- I can't wait to meet Bill.

KEVIN SCOTT: Yeah, it's awesome. So, let's go ahead and chat with Bill.

[MUSIC]

KEVIN SCOTT: So, next up we'll meet with Bill Coughran. Bill is one of my engineering heroes. He's a researcher, teacher, entrepreneur, and business executive whose career includes big jobs like head of the computing sciences research division at Bell Labs, the birthplace of UNIX, the C and C++ programming languages and other technologies that are foundations of our modern tech world. And Bill was SVP of Search, Infrastructure, Maps and Local at Google. Bill is now a partner at Sequoia where he invests in, advises, and coaches tech entrepreneurs.

[MUSIC]

KEVIN SCOTT: The thing that I think everybody will tell you about Bill is he is just by temperament and depth of knowledge one of the best people in the world to work with as an engineer. And everybody who's ever worked for Bill wants to continue to work for Bill as long as they possibly can. I'm thrilled that you are here with us today.

I've been hesitant to ask you to come on because we've had lots and lots of people on the show, you in some ways are the most intimidating one because I admire you so much. (Laughter.)

BILL COUGHRAN: You're very kind, Kevin. Thank you for the invitation. And I also, as I think you and some of the audience probably knows, I don't spend a lot of time doing public things. But with a kind invitation, I'm very pleased to be here.

KEVIN SCOTT: Yeah, and we're delighted. So, I really want to get into your story. So, you got your PhD in computer science --

BILL COUGHRAN: Yes.

KEVIN SCOTT: -- from Stanford.

BILL COUGHRAN: No, I finished my PhD and then I went to Bell Labs. I did do work as a commercial programmer while I was at Caltech, but that was a side job. But I went to Bell Labs and I did a lot of work in fluid modeling and semiconductor modeling early on.

And then became interested in what today would be called distributed computing and kind of shifted to that. And then I was asked at some point to become the leader of a group, and then ended up I stepped up a few steps and ended up as the head of the computing science research center.

KEVIN SCOTT: So how did that transition go for you? I mean, I've always been interested in this because you know I think the people who have worked for you and with you admire both your ability as a like a technical leader as well as your technical ability. And, like, those two things don't always go hand in hand.

BILL COUGHRAN: Some people think they're inversely proportional. (Laughter.) The -- it's interesting, I've always tried to read and explore different technology areas. My sense is, Kevin, you're the same. And so I try to stay conversant. I also try to develop relationships with people that I work with who I believe are experts in the areas that they understand more deeply than I do.

And I just try to synthesize as much as I can, but over the years, I've also learned like it's silly of me to descend into the code in a major project because I can't contribute that -- in that way anymore, but I can think about how does it fit in end to end? What are the constraints it's operating under? You know? There are often big choices about what to do in terms of the basic algorithmic approach and so forth. And I try to stay current with that.

But over the years, it's funny, I shifted away from the numerical modeling stuff, even though it's sort of come back in the form of machine learning and became much more deep on the distributed systems side. But you can learn new things.

One of the things that always amused me when I ended up at Google was I had never worked on a Web service before I joined Google in early 2003. And yet, somehow, over a fairly short period of time, I became responsible for the engineering associated with Google.com. So, you know, it appears old dogs can occasionally learn new tricks, so -- (Laughter.)

KEVIN SCOTT: I think you're being very modest. (Laughter.) But you -- you've done all this work in distributed computing and like you had founded a network company. So, like you had tons of very relevant experience at how to build, like, high-scale systems.

BILL COUGHRAN: Well, so what happened to me, so I started life at Bell Labs very much in the research mold. So, success was writing papers and giving talks and what was a very academic style, success metric.

Over the years at Bell Labs, I felt like I was going to conferences and seeing the same subset of 3- to 500 people over and over again. And I got a little bit frustrated with my ability to impact things.

And so, I guess somewhere along -- during my run at Bell Labs, I decided to get more commercial. So, I took over as a general manager of a business that we started as well as the computing science research center, and I tried to start to think more about what's involved with actually building products and doing sales and so forth. And so, we decided to try to put a Bell Labs in Silicon Valley near Stanford.

And so I came out to do that in late 1998 or 1999 and what I ran into I think is a little bit of what we see today in the Valley, which was it was just a very frothy time. It was before the dot-com bubble burst and you know, people were funding stuff that probably didn't make a lot of sense, but there was a lot of money chasing after ideas and eyeballs at that time.

And I was trying to establish this sort of more traditional research lab. And after trying to do it for a year and a half or so, I threw in the towel and went off with a colleague to found a small startup in the area that was doing a telecommunications system project. And I was the CEO, he was the CTO. And we built that company up from the two of us to about 140 people and then I -- we brought in a CEO to do the go-to-market side.

I left and was going to start another company and then some old Bell Labs colleagues called me and said we needed adult supervision at Google, and that's kind of how I ended up at Google.

KEVIN SCOTT: So, how did you get interested in technology in the first place?

BILL COUGHRAN: So, I'm old enough that I think we call ourselves "baby boomers" now, but at least when I was a kid, we were part of the -- what was then known as the "Sputnik" generation, which were people who -- you know, the U.S. was responding to the space race with the then Soviet Union, and there was a lot of interest in science. And so, from a very early age, I was interested in science and mathematics.

KEVIN SCOTT: Were your parents engineers or technical folks?

BILL COUGHRAN: No one in my family were technologists. It was just something I was attracted to and I was reasonably good at. And so, I went down that path.

KEVIN SCOTT: Where did you -- I'm just sort of curious, where did you even find knowledge back then about how to program? Did you have mentors? Were there people helping you? Were you able to find books in libraries?

BILL COUGHRAN: I mostly found books in libraries and then when I got access when I was in high school, I got access to the local university's 1620 machine. I got to speak and spend some time with some of the people actually doing programming. And so, they acted as mentors to help me.

KEVIN SCOTT: Yeah, I'm, it's -- one of the things that I think is easy for all of us to take for granted now is how not just sophisticated our programming systems are and how much power is at the disposal of your average programmer, but how much of the information about how to learn how to do software development, how much of computer science is all out on the Internet publicly available. There are, like, all sorts of useful videos explaining it from sort of elementary like super high-level, you know, sort of perspectives all the way down to, like, the deep nitty-gritty research.

BILL COUGHRAN: No, I agree. I think the other thing that's changed over the years is access to computing equipment. When I first got access with the first two or three machines, they were machines that, you know, today would have cost the equivalent of hundreds of thousands of dollars. And they weren't machines that were in every elementary school, every high school, every home. And so, there's just -- your exposure and availability to compute -- both learning materials, but also just the mechanical machinery, the actual computing devices is much easier.

Although one thing I have noticed over the years, having been a leader in different groups over the years, I think the knowledge of the actual computing devices has gotten weaker and weaker. And it's been interesting in the last few years as people have become interested in machine learning and high-performance computing again, people are rediscovering all the nitty-gritties of limited memory cache lines -- a lot of the very low-level stuff that I think the early developers, for instance, the people who created Unix thought about how do you fit lots of capability into a very small footprint?

KEVIN SCOTT: Yeah.

BILL COUGHRAN: Both memory-wise and computing-wise, and we've lost a lot of that. And we're rediscovering it.

KEVIN SCOTT: Yeah, and like one of the things that I've always noticed is you know we operate on all of these high-level abstractions to do our jobs. And the engineers that I've found who are sometimes able to do the most extraordinary things are ones who sort of disrespect -- or are able to disrespect the abstraction boundaries, where they can sort of say, okay, this is not working, I don't understand what's going on, I'm not getting the performance I want out of this. Like, there's some weird bug and they just sort of punch through to lower and lower layers.

So one -- like, one of my theories about Jeff Dean, like, he was a compiler guy, right, by his PhD work. And compiler folks they're very good at just getting this end-to-end view of computer systems and like being able to wring extraordinary performance out of things even though, like, they're not actually writing the optimizer bits of the compiler anymore.

BILL COUGHRAN: No, I completely agree with that. Having worked with Jeff and his colleague, Sanjay, they both I think were willing to go very deep into the technology and bypass layers. And some of the people I've known for years, I've worked with before like Ken Thompson were definitely of that ilk as well.

I think one of the things that modern software suffers from is I think people sometimes do not spend enough time thinking about what the right abstractions are and how to make them very simple.

I think one of the things I've admired about Unix, which you can see in Linux today, is a lot of the very low-level system calls that were introduced in Unix have stood the test of now multiple decades because they were very thoughtful about what should be here and what shouldn't, and they didn't add a lot of complexity or cross-functional capabilities. They were all pretty clear interfaces. It's something a lot of modern software designers don't take the time and thought to do it right.

KEVIN SCOTT: And how much of that do you think were the individuals and their tastes and how much of it was just the constraints they were operating under? I mean, like obviously, like a lot of people at the time were operating under the same constraints and not all of them invented Unix. But the constraints have to be useful, right?

BILL COUGHRAN: The constraints are useful, but it is true that we live in a much more complicated world now. I think in the period where some of these early time-sharing systems were built and a lot of the mini-computer work that then led to the personal computer work, you know, people had glass teletypes, they were 24-by-80 character fixed-width kind of devices for output, very simple input, and now everybody expects a lot of graphics, a lot of visuals.

I think just the human-computer interface stuff that we do is much more complicated. You're right, I think the best systems designers know how to think about simplicity and scalability, but our world has continued to get more complicated, so --

KEVIN SCOTT: Yeah, it's like one of the most interesting tensions I think we have in modern software engineering.

BILL COUGHRAN: Agree with that.

KEVIN SCOTT: So, you get access to this machine at a local university, like you continue to get better and better programming. You go to college, you major in computer science?

BILL COUGHRAN: Well, I think like many entering freshmen at Caltech, at least in those days, I wanted to be a physicist. I think probably about 90 percent of the incoming class at that time wanted to be physicists.

KEVIN SCOTT: And was Feynman there at the time?

BILL COUGHRAN: Feynman was there.

KEVIN SCOTT: Okay, so that's a good motivator.

BILL COUGHRAN: Yeah. And I had the opportunity to meet with him a few times in the years I was there, an amazing person. Although, his books are much easier to read than it is to take the exams associated with them. (Laughter.) But that's -- but I think like many of the entering freshmen, I discovered I wasn't smart enough to be a physicist. So, that got me into math.

My first published paper in a refereed journal was actually when I was a freshman -- it was a computational chemistry paper that I did with a post-doc and one of the faculty members. And I had the opportunity to do things like that during the four years I was there.

KEVIN SCOTT: Do you remember what the paper was about?

BILL COUGHRAN: (Laughter.) There's an approximation to a quantum chemistry problem that's called a Hartree-Fock equations, and so this was a simulation on, at that time, a very large IBM 370 or 360-something-or-other that we ran overnight and calculated a bunch of ground states of different small molecules.

KEVIN SCOTT: That's super cool.

BILL COUGHRAN: A long time ago.

KEVIN SCOTT: No, but super cool. You know, and again, like, we'll get back to this later. You already mentioned that you know like we have this sort of resurgence in like all of the things that like people were interested in in high-performance computing back in the like '70s, '80s, and '90s with all of the like large-scale machine learning stuff that we're doing right now.

But you know, I think that the things that you were doing back then are probably like largely transferrable to like things that people are doing now to train machine learning models.

BILL COUGHRAN: Yeah, it's interesting. When I went to Stanford, my primary work as a graduate student was in numerical modeling. My thesis was on numerical techniques for partial differential equations, fluid flow, and things like that.

Did a lot of work with linear algebra and so forth. But it was all early high-performance computing work. And in those days, you know, there were controlled data machines around and Crays were starting – Cray-1 was starting to appear and so forth. And so, there was a lot of interest in different ways to do abstractions in early high-performance computing work.

And it's interesting. All that linear algebra, things that I got familiar with then have been rediscovered, of course, as part of the machine learning revolution that we're having today.

KEVIN SCOTT: Yeah, I mean, it's really, really fascinating like as I spent a bunch of my time trying to figure out like what the infrastructure needs to look like to support training very large-scale models. I, in a very small way, had been doing work when I was in grad school in high-performance computing. And it's just remarkable to me.

Like, I thought, you know, funny enough when I took my job at Google in 2003 that all of the stuff that I'd done in graduate school was like gone forever and like I would never -- I'd never see the day where it was useful again. And then I'm, you know, sitting here in 2019, and surprise, surprise. Relevant. (Laughter.)

BILL COUGHRAN: No, it's true. And, of course, high-performance computing developed hybrid machines, (inaudible) architectures, all kinds of things that are reappearing now in the machine learning context, so --

KEVIN SCOTT: Yeah, well, you know, it's sort of interesting, right? And I'd love to get your take on this. We're at this moment now where there are certain flavors of machine learning, so this simulation-based reinforcement learning stuff that has achieved some fairly spectacular results that have been publicly disclosed over the past few years, like mostly in game playing, and then there's like recent stuff and natural language processing on unsupervised learning.

And in both of these cases, like the workloads are just because you have gotten human supervision out of the direct training loop, like you don't have to produce a bunch of label training data, like you're sort of -- reinforcement learning, you're gathering all of your data to train the model inside of a simulation environment, and the unsupervised learning, like, you just have like a large, like, Internet corpus of data that you're training on and you're trying to like induce some structure out of it.

So, they're like absolutely insatiable in terms of their compute appetite. I mean, it's just mind boggling and like at least with the unsupervised models, and I think the same thing is true with these reinforcement learning models, the more compute that you can throw at them, like the better the result you get. And so, like there's just this real tangible incentive to like throw them more compute.

And it's happening at exactly the same time where Moore's Law is running out of steam. And it's not Moore's Law that's the problem, it's sort of the Dennard scaling that's the problem. Like, we can put, you know, tens of billions of transistors on a die now, we just can't power them all.

BILL COUGHRAN: Correct.

KEVIN SCOTT: And so like you've -- like, we really are having to like revisit computer architecture and, you know, like, how do you cool these things? How do you design their networks? Like, what is the memory hierarchy? I mean, it's just crazy that all of this is relevant again.

BILL COUGHRAN: No, I agree. And we're having, obviously, a revolution again in computer architecture for all the reasons you point out. And you know, John Hennessy and Patterson just won the Turing Award for basically having thought about computer architectures in different ways, done RISC, written their book, and kind of I think they reopened the door as part of their Turing Award speech and some of the speeches they've given since, suggesting it's time to rethink computer architecture. So --

KEVIN SCOTT: Yeah, I completely agree with them.

BILL COUGHRAN: The other thing that's been interesting with the insatiable appetite for computing, and it worries me a little bit, I still believe that universities are the best training ground for people doing computer science work. You know, maybe not everybody should take the time to go get a PhD, but I think having a rigorous program with lots of mathematics and theory and so forth and not just -- it's more than just programming, but I think one of the challenges with the rise this time of machine learning is I think the universities are at a disadvantage.

KEVIN SCOTT: Yes.

BILL COUGHRAN: Because, you know, the big platform companies which of course would include Microsoft, Facebook, Google, Amazon, and others, are the best place to get your hands on lots of data and lots of computing resources. I think it's very hard for MIT or Caltech or you know Stanford to compete with those capabilities.

And so, I think it does call into question what's the right partnership between universities and these large companies? Because I think otherwise we're going to have a lot of people who are deficient in their training as they enter the workforce and try to do novel things.

KEVIN SCOTT: I completely agree. And it's become especially obvious I think over the past year where some of this stuff on like the very frontier of AI, like the computations have become so large and it's so expensive that we even sort of have to think carefully about it inside of like these big tech companies. Like, you just can't have 5,000 people all like doing each a $10-million computation to train their model. Like, you have to sort of think about like how you want to focus these efforts in ways where you can, you know, where you can get leverage out of your compute infrastructure.

And I think it's especially hard for these universities. I've joined the advisory board of the Stanford Human-Centered AI Institute. And thinking in general about like how we can form these partnerships where we can make resources available.

I think it's beyond just training, though. You also really need you know sort of dispassionate third parties, you know pushing in interesting new directions. Like, one of the things I really worry about with the state that we're in in machine learning right now is that we have made a bunch of subjective decisions about how these computations should be performed by virtue of the frameworks that we've built.

And so like it's sort of like MapReduce, right, back in the day. So, as soon as you have MapReduce, it's like such a powerful abstraction mechanism that it solves like a ton of different problems and like abstracts away like all of this tedium from you know like how to you know massively you know in distribution fashion run these workloads.

And, you know, the tool is simple enough and powerful enough where all of a sudden like everybody wants to try to bend their computation around where it's like a MapReduce job. And I think we have a little bit of that right now. I mean, so like you look at these deep learning algorithms, like, whether it's TensorFlow or PyTorch or like the numerical kernel of all of these things is stochastic gradient descent.

And, like, because you've chosen SGD as the way that you're going to like do the fitting for the models, it just sort of excludes a whole bunch of things that you might want to do. And so, like we're -- in a real sense, we have been pushing people down to this like what has to be a local -- like a local optima. And you know, I would love to have like just a much, much larger pool of people sort of pushing at those assumptions. But in order to do it and to compete to show that the new thing is as effective as the, you know, the old thing, like, they have to have the compute resources.

BILL COUGHRAN: Oh, I agree. And I think the other thing which, of course, has happened more and more with machine learning is not only are we constrained by the frameworks, but we're also I think even in the unlabeled case, the case where you're not pre-labeling data and you're doing unsupervised learning, I think you have the risk that if you feed it the wrong sets of data, you can generate biases of various kinds without even realizing it. And it's tricky. There have been a number of published results on this stuff, but it's kind of insidious.

KEVIN SCOTT: Yeah. So, let's go back. So, tell me a little bit more about this experience with starting a company because that is a very unique thing. It is not at all like managing researchers, it's not like -- you know, even though Google hadn't gone public when you got there, like it was already a reasonably big thing.

BILL COUGHRAN: Yeah, it was 4- or 500 people.

KEVIN SCOTT: But like zero to 140 is like -- I mean, that is a -- that's something else.

BILL COUGHRAN: Well, you learn to tell stories to venture capitalists. So, now that I'm doing that as a full-time job, you get to play the other side of that drama. But you have to think about what's the product you want to build? What kind of team do you need? How much funding do you need? You have to think about everything. And starting as you say from zero, you've got the challenge of, you know, you're the janitor when day one starts, right?

And so, you have to think about everything -- you'll laugh. I did the payroll and the bookkeeping for the first few, I think, first few months, not that I'm particularly good at that, but I can do a little bit of accounting math.

KEVIN SCOTT: Well, and it must have been good preparation for what you ultimately had to do at Google because at some point you weren't just an engineering director of like a group of people, you were sort of like the CEO of like an army of engineering people. And so, like you had HR issues, you had finance things, you had facilities things. I mean, like I remember walking into your office at one point, you know, where you were, you know, sort of looking particularly exhausted.

And I was, like, you know, what's going on? And like you had this spreadsheet in front of you like where you were thing to like figure out where everybody was going to sit who was being hired in the next unit of time, which is probably very far away from writing papers about, you know, ground state simulations of quantum chemistry systems. (Laughter.)

BILL COUGHRAN: No, it's true. I, well, so I think the role I fell into at Google was I was willing to take on stuff that not everybody else was willing to take on. So, there was a small executive committee that ran Google, of which I was a part for a few years -- number of years. And I -- for a while, worried about things like facilities and HR. And you have to do this in a young company. You have to do what you need to do to make it successful.

And part of the challenge and you saw this at Google, I'm sure you also saw it in your experience at LinkedIn, it sounds very hard to go from zero to 140, it's also very hard to go from a few hundred to several thousand. The scaling, trying to figure out what works. You want to introduce more process and structure, but if you introduce too much, it kind of kills all the energy and excitement in the team and so forth. And so finding a path through a heavy growth phase of a company is pretty tricky. And I did a lot of that for Google in the early years.

KEVIN SCOTT: What do you think was the most challenging thing that you -- or problem that you had to solve in those early years?

BILL COUGHRAN: I didn't join Google until early 2003, but I think from when the dot-com bubble burst to probably 2004 or '05, recruiting was hard, but it wasn't crazy like it is today because there weren't enough kind of growing, powerful new companies forming because of the sort of downdraft that happened because of the dot-com burst.

I think today, one of the challenges I see with the young companies I work with is just figuring out where to recruit and what to do. One of the things you see from more and more companies is looking at doing engineering away from Silicon Valley.

KEVIN SCOTT: Yeah.

BILL COUGHRAN: And then I think one of the biggest challenges I saw in the early days at Google was it was still very much a startup, even though when I joined it had a few hundred employees. And there were key areas inside the Search product, for example, where only one or two people actually knew some of the key things to get things done.

And so one of the challenges that I found particularly hard was figuring out a way to not disempower and take responsibility away from the people who had been standing in the breach, so to speak, but building a team around them so that you could actually have turnover and have organizational stability, but you sort of transition from a few kind of superhero type people that know all these key things to teams that actually own things.

And that's an enormous cultural change, which we went through. I think we managed to preserve a lot of what was good about the original culture, but it was hard.

KEVIN SCOTT: Yeah, another, I mean, just sort of outside looking in, like one of the things that you always seem to have a real talent for was managing the very senior, like, very most capable engineers and like figuring out how to you know let -- give them the space to do their best work without like sort of having the crush of like all of the sort of outside world and you know sort of friction that can sort of land on you as an engineer.

I mean, like, at one point, you had like almost all of or maybe all of the principal distinguished engineers and fellows reporting into you. And like they're not an easy group of folks to manage. I mean, I know many of them, I love them. Like, they're like fantastic human beings. But they're smart, they're -- they have strong opinions, they're you know sometimes their interactions you know with one another or like with other engineers are sort of like interesting.

But somehow or another, like, you built this environment where all of these folks could like really flourish because at some point, like, they just don't -- they don't have to work for you, they don't have to work at Google, they can do anything that they want to do.

BILL COUGHRAN: So, I think what I tried to do, and I think I was successful more often than I wasn't, was find areas that were important to the company that different individuals were interested in. That's how we -- you know, we spawned a project called Borg internally to do cluster management. There were a group of people -- if you look at the work that was done there, it was very good, it was very deep, it was ahead of its time.

But it also was rediscovering mainframe timesharing, but on a cluster. And so some of it was recycling ideas that were prevalent in the '60s and in the early '70s, but finding the right set of people that were interested in that problem, were motivated by the problem, and putting them together. So, finding things that were interesting, you know, we had the challenge of managing locking and global namespace stuff, and that was an area that Mike Burrows was particularly attracted to and did a terrific job with. And I think so it was just picking individuals. But it didn't always work.

KEVIN SCOTT: That particular one is sort of genius like sort of saying like here's this person who can work on these problems and like we're going to go build infrastructure that everybody else can use. I mean, those are some of the hardest problems in software engineering. Like, proving that you know, Paxos is implemented correctly in like one of these locking systems. And like Mike is, you know, maybe the best person in the world, like, one of the best persons in the world.

BILL COUGHRAN: No, I agree. And Google had -- and still has people that are absolutely world class and strong. But you're right, some of them don't mix well together. You have to figure out the right problems, and it didn't always work.

The other thing I often tell young companies that I work with now is Google also could cheat. And I don't mean that in a nefarious way, with the advent of AdWords, it started to become you know lots of positive cash flow--it gave flexibility -- the fact that the business was so strong gave us access to resources and capabilities we wouldn't have had. And one of the things I always find amusing is when I am working with a very young company and they want to talk about creating 20-percent time.

And I often will tell them when you have a massively profitable business, why don't you come back and ask me about 20-percent time. So, I think people need to be more realistic about what constraints you're operating under.

And at Google, one of the things I did is like we started storage projects we weren't absolutely sure we would need because we had the flexibility and at another company you wouldn't do that. Most of those bets turned out to be good bets, but not all of them.

KEVIN SCOTT: Yeah, well, you know, it's sort of interesting, like even 20-percent time, like there are things like that that take on these mythological proportions. And, like, I don't ever remember using 20-percent time. And it wasn't because I didn't think it was available, it wasn't because my manager wouldn't have supported it, it was because I was so interested in the work that was my 100-percent time that I didn't want to stop doing that to go do this other thing. I mean, it was nice to know that I had it if I wanted to use it, but I think a whole lot of people like –

BILL COUGHRAN: Most people did not use it. Yeah. And not everybody had an exciting new project they wanted to start on their own, so --

KEVIN SCOTT: The thing that I really enjoyed out of Google as it was growing as fast as it was, so I was there like I joined I guess a few months after you did in 2003 and then I left the first time in the middle of 2007.

And, you know, the thing that was just extraordinary for me over that period of time is like because we were growing so fast and we were solving so many problems for ourselves the first time, you got to be this sort of participant in the unfolding of like this really interesting great business and this really interesting great technology stack. And you sort of got to see, okay, this worked, this didn't work.

And like in a very quick period of time, like I felt like I got exposed to like a whole bunch of things that I never would have been able to get exposure to if I had been at some place that wasn't growing that fast.

BILL COUGHRAN: Oh, I think that's right. I agree.

KEVIN SCOTT: Like, when I left to go do AdMob, which was already a company I joined in 2007 to run the engineering team when it was like -- the whole company was about 25 people I think.

And like I felt just because I'd been able to raise my hand and say, oh, yeah, like I'll go help with the mentorship program. I'll go help set up a faculty summit. I will go like you need someone to go due diligence on like an M&A thing that we're doing. Like, none of which I was qualified to do when I raised my hand, but because no one else was there who was raising their hand to do it, like I got all of this experience.

And when I went to AdMob, it was like, okay, like, I'm like there's still a whole bunch of stuff I need to learn, but at least I'm confident in like a big enough set of things where I can, you know, I can have some modicum of comfort that I'm not going to screw everything up. (Laughter.)

BILL COUGHRAN: Yeah, you can learn a lot. And you get a lot of advancement and opportunities to do different things, right? It's a special time in many companies, so –

KEVIN SCOTT: So, you were at Google for how long? You left in 2011? So you were there for --

BILL COUGHRAN: Eight and a half years I think, yeah.

KEVIN SCOTT: And so how -- what was the biggest team that you had? So, you were up to 5-, 6,000 people?

BILL COUGHRAN: Yeah, it was probably a little bit north of 6,000. Google broke itself up into product divisions, they call product areas. And I think at one point, I had amassed what are today four or five of their product areas.

KEVIN SCOTT: So, let's talk about the transition to venture capital. So, you've been doing this for I guess almost eight years now.

BILL COUGHRAN: Close to eight years.

KEVIN SCOTT: And so like what made you could have done anything. You could have stayed at Google, you could have like gone to another company and done a similar thing, you could have started a company. You could have retired. You could have basically done whatever.

BILL COUGHRAN: Yeah, I think when I look back at my career, so I thought -- one of the challenges of the role that I had at Google is I was unsure what I could do at another company that wouldn't feel like a step down. And it's not that I'm in love with a particular title or a particular role, but there aren't a lot of Googles in the growth phase out there. I think there are probably a small number, but they're very few and far between.

And so, I didn't see how to repeat that. And I think one of the things I've gotten better at over the last ten or 15 years is mentoring and spending time with people and helping them think through their problems and their challenges and so forth.

And it felt like I could do that as a board member, as a mentor to a bunch of young companies. And so for me, some of it's what people I think call "giving back," but some of it is I think at this stage of my career I'm probably better as a coach than as a player.

KEVIN SCOTT: Yeah, well, I mean, I remember some of the best advice that I've ever received has been from you. And I remember like just vividly this one like sort of critical moment in my own career. So, this is when I was thinking about leaving Google to go to AdMob, I'd chatted with a few companies, like I had chatted with Omar, who's I guess now one of your partners at Sequoia, who was like the founder and CEO of AdMob.

And I remember I asked you for a meeting, which you were always generous enough to take when you were much, much busier than I was. And I remember coming into the room and I was like very reluctantly admitting that it's like, oh, I've been out talking to these companies and like I'm thinking about leaving and you already knew because somebody had -- whose name I will not name had -- (laughter)

BILL COUGHRAN: Yes, I think I know who, yes.

KEVIN SCOTT: And, you know, like done a background check, I'm guessing. But you gave me the best advice. You were like, okay, like, here's the advice I'm going to give you with my Google hat on, which is you know, whatever that was.

And then here's, you know, the advice I'm going to give you without the Google hat on, which was like just super helpful to have both of those perspectives in making that decision, which was really hard. Like I had a hard time leaving because it was really -- it was a great company and I had so many friends there and I felt so grateful for the opportunity that I had gotten, which I think a lot of it was luck. There's no way that --

BILL COUGHRAN: There's a lot of luck in many outcomes in life, I've decided. (Laughter.)

KEVIN SCOTT: There's a lot of luck. Just extraordinary amounts of it. But, you know, like that advice you gave me was like one of the -- and it was certainly like a pivotal moment, like the whole -- my experience with AdMob, like without that, I wouldn't have been able to you know get the role that I had at LinkedIn. Without the role at LinkedIn, I wouldn't be sitting here as CTO of Microsoft. And so, you know, like in -- so, you know, I'm glad that you're out there like giving other people this like really good advice.

So like, what made you decide that like this is the sort of thing that's worth your time? Because you were doing it even before it was like officially your job as a like a venture capitalist. You didn't have to take that meeting that day or like any of the other times, you know, because I was way below you in the org. And you weren't just doing it for me, you were doing it for a bunch of other people.

BILL COUGHRAN: So, maybe this is one of the things I've tried to do as a leader, I think most people -- you know, when you're employed by a company, you're employed by Google, now I'm employed by Sequoia Capital, you have to have loyalty to your employer, but I also think as a leader inside of any organization, you also have to be authentic. And you have to -- if somebody asks you for advice, they don't want the company line. They want real advice.

And so what I've always done in the last many years is do exactly what you described, which is I will give you my advice as a leader inside a company, but then I will also give you my advice as an individual. And I think, to me, that's been one of the things that I think has bonded people to me over the years, because they feel like they're not getting a bunch of political mumbo-jumbo, which unfortunately I think particularly folks in large corporations, I've known a number, they can only talk in the sort of party line mode and they never get out of it.

And it makes them appear, even if it's not true, as phony. And so I think part of it's just being genuine and authentic when you talk to people.

KEVIN SCOTT: It's sort of scary to do because when I was running engineering and operations and LinkedIn, which grew over the course of the six years that I was doing the job from a couple hundred people to about 3,000. Like I took this queue from you that I was going to try to be as authentic as possible when someone came to me and like they were struggling with a problem and like trying to figure out like what the next step was for them.

And like I would do almost exactly the same thing that you did for me. Like if they were contemplating a job offer, it's like here are all the many reasons why I would like you to stay like because we really need you here and, you know, like we walk through the whole thing.

And then I would give them, you know, my unsolicited advice, and sometimes it would be like, hey, I don't think that this is like based on what I know about you and what you've told me about your aspirations, like I don't think that this is a good move for you. And sometimes I would tell people, it's like this isn't a bad option. Like, you are now faced with a tough choice, like I think it's a -- you know, you've got a good option here and you've got a good option there.

And the thing that I found, though, like even it's a horrifying thing like when you are leading you know a group of people and like you and in a very sort of very concrete way are dependent on them to like be able to achieve all of the things that, you know, you're trying to get your company to achieve.

But I found like having those conversations like I think because people believe that I had a genuine interest in like their success and their happiness, like I wound up being able to hold onto a bunch of people like way longer than I would have otherwise.

BILL COUGHRAN: I think that's right.

KEVIN SCOTT: But it was hard to do because it feels very scary.

BILL COUGHRAN: Yeah, it's risky. It feels risky. I understand. I think the other thing which I saw at Google and I suspect you've seen in your arc, too, is Google hired a lot of very bright kind of new grads or it was their first or second job, and it was fairly common for people after a few years to kind of wonder what the world looked like outside of the particular confines of Mountain View at the time. And that's not irrational for a young person.

KEVIN SCOTT: No.

BILL COUGHRAN: And on the other hand, Google wanted to retain those people. And there were some difficult challenges in that.

KEVIN SCOTT: So, what's -- what are the most interesting things that you're seeing in venture now? So, like, it's -- it is an exciting time right now because like, there's a lot of venture capital out there. It's never been easier to start a technology company because you have open source, you have all this cloud infrastructure, you have like an increasingly powerful set of capabilities that let small teams accomplish big things.

And like you're at Sequoia, which is like one of the top venture capital firms in the world. So, like, what's interesting right now?

BILL COUGHRAN: Well, there's clearly a lot of interest in software around machine learning. I think that there's a bunch of companies trying to create tooling or do vertical applications.

I think there aren't very many companies proposing to do platforms because you've got TensorFlow, PyTorch, and other things in the market from companies that have, obviously, much larger resources than you can mount from a startup.

But the sort of machine learning software and getting machine learning that's usable and digestible by a broad community is very hard to do. And I think there are some interesting ideas and companies around that.

There's a lot of interest in processor technology both for machine learning, but also I'm starting to see some companies that are talking about rethinking more server processors more generally and so forth. And so, I think there's a lot of that.

Those are harder companies I think for venture capital just because it requires hundreds and millions of dollars of investment rather than you know tens of millions of dollars of investment. So, I think -- but I see interest in that area.

There's a lot of interest in quantum computing, which is -- you know, people will debate whether it's about to go commercial or whether it's still going to be research for a few more years. It feels to me like we're on the cusp of a bunch of things right now in that area. Those are very interesting.

And then there are interesting companies in storage and networking and so forth, but I think the big things have been processors, machine learning, and then quantum are areas that I think are particularly interesting.

KEVIN SCOTT: And the processor trend, aside from the machine learning stuff, so like we're seeing with distributed systems, right now that like architecturally, like one way to compose these things now is streams and stream processors, which are going to have different compute requirements than normal sorts of processors.

There's also like this interesting thing where you sort of reach your sort of power saturation point with transistors for logic, but you haven't yet for memory cells.

BILL COUGHRAN: Yeah, I think there's tradeoffs now that people are rethinking about -- between you know actual logic-type processing versus memory, what kind of embedded memory do you want, what kind of memory structures, you know, there's obviously new kinds of memory coming into the market that don't require DRAM refreshes, but are faster, that's another area that may be interesting.

I think the other thing which strikes me as a big challenge as people think about processors themselves is you know, there's been some fascinating work on the security side and very scary, people did not think through all the risks with speculative execution. And on the other hand, if you throw speculative execution out, you hurt performance a lot.

And so I think there are people trying to think about how do you think more from first principles about what the security risks are in processor design so that -- because when you admit security weaknesses at the very bottom layer of the hardware, it's pretty hard to defend against.

KEVIN SCOTT: Yeah.

BILL COUGHRAN: And it's scary in new ways, right? And so, I think there are a lot of interesting things happening in and around that area.

KEVIN SCOTT: Yeah. Are you seeing the stuff with the -- in the IoT space? Because processors there like still have a couple of generations of, you know, real Moore's Law left in them.

BILL COUGHRAN: So, we're seeing a bunch of companies around -- there are a lot of people looking at processors for the edge where they can do some inference, maybe a little bit of training I think, and very low-cost kind of microcontroller-level designs and so forth.

There's a number of interesting new ideas and companies in that area. I'm -- when I've looked at IoT, I've mostly looked at things related to industrial IoT because I think consumer IoT has been tricky for a variety of reasons. One is, it's often driven by brand and consumer reach, which I think advantages companies like Apple and others. And I think the other thing is smart phones are too good at doing a lot of different things, and so the killer apps for the consumer IoT I think has been pretty limited so far. Personal opinion, but there you go.

KEVIN SCOTT: Yeah, I mean, we can certainly see that the industrial IoT stuff, like the applications there are like rich and like there's an enormous amount of opportunity there. I think we sort of see the world in similar ways.

BILL COUGHRAN: Yeah, no it's -- consumer IoT I think more and more devices are connected and have some ability to respond to command and so forth. And we'll see a proliferation of that, but special use cases I think are -- there aren't enough interesting use cases.

I talked to people that have Alexa devices or you know the Google Home devices and so forth and they use them to set alarms and timers and play certain kinds of music and so forth, and then they rapidly seem to run out of other things do with them.

KEVIN SCOTT: The thing that I think is a little bit interesting, though, is that the capability in these devices already like in both the hardware and the software is much higher than what we're using it for.

BILL COUGHRAN: Absolutely. (Laughter.)

KEVIN SCOTT: So, you know, like just as a proof of concept thing, I'm building an AI coffee machine right now, which probably sounds weirder than it actually is. But like the idea is, can you build a consumer appliance from scratch, like as an individual, although like you know, one with slightly more resources than an average individual.

But, like, can you build a device that has like a modern AI-powered user experience in it? And so, like this thing will have no buttons, no displays. And it will have a camera, a speaker, and a microphone. And so like when you walk up to it and like it knows that you're paying attention to it using very straightforward computer vision models, it will say to you, "Can I make you a cup of coffee?" And it will be able to recognize who you are and remember your preferences and --

BILL COUGHRAN: Yeah, no, that's true.

KEVIN SCOTT: And, like, that's an entirely -- you know, entirely doable mode of like building a user interface on a consumer device right now and it's going to be not too terribly expensive in the not-too-distant future. Like, I'm going to be able to do this with 30 bucks' worth of electronics and like the price of that's going down to two over the next few years, I would imagine.

BILL COUGHRAN: I think you'll see that in a lot of consumer devices, but they'll be very limited and narrow use cases. So --

KEVIN SCOTT: Yeah. I just -- I'm just super excited about (laughter) about what's possible there, so I want to get people inspired to do more.

BILL COUGHRAN: I think human interfaces that are voice driven and have some aspect of computer vision is clearly where things are going. So --

KEVIN SCOTT: Yeah, awesome. Well, thank you so much for taking the time to talk with us today, Bill. It's been a pleasure.

BILL COUGHRAN: It's been my pleasure to be here. Thank you for again for the invitation.

KEVIN SCOTT: Awesome.

CHRISTINA WARREN: Well, we hope you enjoyed Kevin's interview with Bill Coughran.

So, Kevin, one of the really interesting things that you and Bill were talking about is what was old is kind of new again, and kind of how the tech that was -- the problems that we were trying to solve in the '70s and '80s and even the '90s are now kind of coming back in vogue again. Why do you think that it is that those problems are so relevant and that we're still going back and kind of revisiting some of those ideas?

KEVIN SCOTT: Yeah, it's amazing to me how cyclical technology is as a business. Like I've been coding since I'm 12, I'm 47 now, so, you know, 35 years-ish of like paying very close attention to what's happening in computing.

And like just inside of like the 35 years like there have been multiple cycles where you'll like have this intense enthusiasm for a thing and then it will sort of fall a bit out of fashion, and then the next thing comes in and you'll sort of forget temporarily all about the old stuff.

You know, I think these older technologies that pop back up, it's almost a puzzle why they went out of fashion in the first place. You know, it's like you look high performance computing for instance, we were building in the '70s, '80s, and '90s like these very sort of idiosyncratic supercomputers and very high performance machines that all were different in little ways.

So, like they had a good idea about how do the memory hierarchy or a good idea about how to do networking or a good idea about how to split the computation up across a bunch of different processors.

And the thing that caused them to go out of fashion is like there was this sort of catalyst, you know, driven by the rapid growth of the Internet to sort of standardize your compute architectures on commodity CPUs, commodity memory, commodity networks, like commodity everything. And so, like you just sort of created at massive scale like large, large amounts of compute where all of the like individual units of compute looked exactly the same and were built as cheaply as possible.

And like that works really well for, you know, sort of the problems that we had for the first 20 years or so of like the commercial explosion of the Internet.

What we're seeing right now is like the problems are different again, but like different in the same way that they were like a bunch of years ago. And so, like now we are -- we're digging back up all of that good work that folks did a few decades ago and are realizing like how valuable it still is in these new contexts.

It's really fun, because I spent a bunch of the early parts of my career thinking about these things. Like I did an internship at the National Center for Supercomputing Applications like where we like when I was there we had just taken delivery of this machine called the CM5 from a company called Thinking Machines, founded by, you know, a famous computer scientist, Danny Hillis, who like I hope we will get to come on this show at some point. Like Danny is like a super awesome human being and like just like maybe the most brilliant mad scientist I've ever met. I love Danny.

But anyway, you know, this machine was just epically big and like epically cool and just sort of like it was -- it was a challenge to program this thing, but like when you could harness all of its power, it could do miraculous things.

And, you know, I never in my life would have thought that all of this stuff that I did back in the '90s was going to be relevant again one day. So, it's like really I'm sort of glad that technology has these cycles, because it's almost like I get to do two things at once, sort of be nostalgic about like all of this cool old stuff and like I get a set of techniques to solve a set of problems that are very relevant today.

CHRISTINA WARREN: I love it. No, it's so cool to be able to revisit the past but also use the advantages that we have now in solving those problems. That's great.

KEVIN SCOTT: And the cycles are -- like can be really long. Like the other thing, too, like I think is worth thinking about AI right now and some of the disruptive effects that it could potentially have, like we're learning lessons even from the Industrial Revolution, which is like not just a couple of decades ago, it's like centuries ago.

CHRISTINA WARREN: No, that's so true, I hadn't even thought about, but you're right, I mean, these cycles can be really long and we can learn lessons from them -- or hopefully we can, right?

KEVIN SCOTT: Yeah. So, the message for the young ones is like pay attention to history, very, very important.

CHRISTINA WARREN: Okay, so we need to wrap up, but before we do, remember to tell your friends, your colleagues, your neighbors, your barista, your Lyft driver, your dental hygienist -- real talk, I actually have told my dental hygienist and my dentist, I have.

Okay, you know, you get it, tell everybody you know about the show and you can write to us anytime at BehindTheTech@microsoft.com, to give us your feedback and tell us what you'd like to hear more of.

KEVIN SCOTT: Yeah, we really would like to hear from you. See you next time.

[MUSIC]