Talks at GS
Jeff Lawson
Co-founder and CEO, Twilio
George Lee, Moderator
Recorded: December 13, 2021

Jeff Lawson: Experimentation is the prerequisite to innovation. The more ideas you can try, the faster you can try them, the cheaper those experiments are, the more innovation you're going to get.

George Lee: Hi, everyone, and welcome to Talks at GS. I'm George Lee, our co-chief information officer, and we are very pleased to welcome Jeff Lawson, co-founder and CEO of Twilio and author of the new book *Ask Your Developer: How to Harness the Power of Software Developers and Win in the 21st Century.* Jeff, welcome and thank you for joining us.

Jeff Lawson: Thank you so much, George. It's great to be here today.

George Lee: You have founded a number of startups. You've been in the technology industry for a while. Why do you think you got the startup and technology bug so early?

Jeff Lawson: You know, I've always just loved building things, really. And even going back to my very early days, like, when I was a kid, I would build things with my dad, you know, when I was, like, five years old, right? I'd say, "Dad, let's do a project." And he'd say, "Let's make a robot." Okay. And we'd get a cardboard box, and we'd draw on it the various features. And so it wasn't until I got a computer that you could actually make things that really worked.

We got an Apple IIe when I was a kid. And now you can write programs and actually tell a computer what to do and it actually did it, which was very cool. Except at the time, you know, you wrote a program on an Apple IIe and, yeah, you could show it off to your parents or a few friends and they'd be like, "That's nice." Nobody could really, however, experience the thing that you built.

And then came the Internet. And I started college in '95, like, literally a month after the Netscape IPO when it was becoming apparent that this Internet thing was going to be big, right? And the amazing thing about the Internet was you could now make things with the computer. You could write a program that told the computer to do something,

and it actually did it. And you had at that point millions of people could actually experience and use the thing that you built.

And now of course that seems quaint because it's billions of people. But the whole idea that you can build something that actually works, that does something, a utility for somebody else. And if you built the right thing, the Internet provides you with an audience of millions or billions now of people who can use it. That is such an enticing idea because, in the history of human creation, we have never had the ability for one human or a small group of humans to impact so many.

You know, think about it. Think about the act of writing code. You sit down at your computer. You type some magical codes into the computer, into a text editor. And you hit Publish, and it goes in. Whatever you built goes online into the web or into an app store if it's a mobile app. And suddenly overnight billions of people can use what you built. In the history of human creation, that's never existed. For most of humanity, if you made a thing, you made a chair, it's like, okay, dozens of people might sit in that chair and that's what it meant to create something.

But now suddenly just a person with something as simple as a text editor can change the world. To me, that's a calling. What can you do with that?

George Lee: Let's talk a little bit about your time at Amazon. I heard that that job was so secret at Amazon that they really wouldn't even tell you what you were hired to do. How did you ultimately learn about the mission, and what were sort of the early steps that you took in that domain?

Jeff Lawson: I decided that I wanted to get some experience at a larger company that seemed like a well-run, interesting company to learn, like, if I'm going to build a successful company one day, what should I copy? What should I avoid? How would I build my company? And I wanted to learn.

I had a short list of companies that felt like they fit my criteria. Amazon was on the list. And I was interviewed. And I remember, I was interviewing for this role at Amazon Web Services, and I did the whole day of interviews up in Seattle. And Andy Jassy was actually the bar raiser on my interview loop. It's the end of the day. You get to that

point in the interview where the interviewer says, “What questions do you have for me about the job?” And I said, “Yeah, I have a question, Andy. What is Amazon Web Services?” And he said, “You know, I'd love to tell you but I can't. I can't tell you what it is. But trust me, it's cool.”

And as it turns out, that is a fantastic recruiting tactic for a 20-something young person who's very mobile and can just decide to do whatever they want. And I was like, hmm, “You know, I could tell you but I'd have to kill you,” is a pretty good recruiting tactic. So I decided to trust him, and I moved up to Seattle and joined. And my first day at AWS I met my office mate and he was product manager for S3, which was a team of, like, four people at the time trying to invent the future of storage.

George Lee: Amazing.

Jeff Lawson: And I got attached to a product in the payment space. And we had these two people in South Africa building the thing. And just this whole idea that the core building blocks of the things the developers need to build Internet skill applications could be bought for pennies per use and could be a self-service flow. Like, you

know, an Amazon checkout. You just go buy it. You put it on your credit card and you buy it. That was such a mind-blowing idea at that moment in time. This is 2004. The idea of what could you build if you had Internet-scale infrastructure available to you as a developer and all the little pieces that you needed to build an application were just there ready to go?

So it was an amazing time that I spent at AWS. I spent a couple years building out a number of things. And when I decided to leave, it was because I decided it was time for me to start to go back to my entrepreneurial roots, having learned a lot at Amazon.

George Lee: How can you kind of explain what Twilio does and why it's become so ubiquitous and important in the world?

Jeff Lawson: When I left Amazon, I was looking at a bunch of ideas for what I wanted to start next. And I realized that my three prior startups, every one of those companies, despite the fact that they were very different companies, we were using software to go build something better for our customers than what existed in the market at

the time. And the way we did that was by using software really intelligently. So that was the first common thread among all three of my companies.

The second, however, was that, in building those companies, building those products, those customer experiences, we needed to engage with our customers, right? We needed to build communications into one of those many touch points a company has with its customer where, if you communicate better, if you engage with your customer better, they perceive it as a better experience and they're more happy with your products and services.

And so sometimes it was during our marketing or our sales when we were just meeting a customer and starting the relationship. Sometimes it was while they were using our product. Sometimes it was in a service or support experience or even out in the field. Hey, we're trying to coordinate a delivery person and a buyer of a ticket to hand off the ticket to get into the event. That was a StubHub example. But there were all these places in designing the product experience where we needed communications. Yet every time we had this problem, we would say, “Oh, that's a really neat idea, but we're software developers, right?” We

have no idea how that stuff works. Communications? You know, we go talk to carriers. We go talk to the hardware companies like Cisco and say, “You know, we have this idea for an experience we're trying to create. Can you help with that?” And every time, we got a pretty similar answer. They would say, “Yeah, we can help you with that. First step is we're going to roll out all these copper wires from a carrier to your data center. Then we're going to rack up a bunch of hardware. Then you got to go buy this software stack that sits on top of it. And then it doesn't work out of the box, so you got to bring in all these consultants to come. And we think we can bang it all into submission and build this idea you have. Sign here. It'll be $3 million, and it'll take us two and a half years to build it. But we can get started.”

And every time, I remember having the same response. It was, like, whoa, first of all, the money, like, millions of dollars. The more offensive thing than the money actually was the time.

George Lee: Was the time.

Jeff Lawson: And so we started Twilio to solve that

problem and make it so, when they had these ideas for, like, “Oh, when this happens, I want to be able to text my customer. Or if a customer calls me, I want to be able to automate their access to all this information. Or if I want to build these complex email workflows, that's just plugging in some APIs.” We wanted to put all of those types of experiences into the tool belt of every software developer in the world.

George Lee: I mean, it's such an interesting through line of what you've described even thus far in the talk. That sort of awakening yourself to the power of software. Then working at Amazon and seeing the power of abstracting complexity to something that developers can use that makes it much simpler, more available. And then with Twilio, taking that really to a feature level, which is again that desire of or that instinct to create extraction levels and make it that simple for developers to do their work and be creative.

Jeff Lawson: Yeah. Well, you know what? It's interesting. Experimentation is the prerequisite to innovation. The more ideas you can try, the faster you can try them, the cheaper those experiments are, the more innovation you're

going to get. And I think back, I was very moved. Several years ago, I read this book. It was a biography of the Wright Brothers.

George Lee: Ah, great book. Great book.

Jeff Lawson: Yeah, have you read that?

George Lee: Yeah.

Jeff Lawson: And it was amazing because you read the story of these two bicycle mechanics from Ohio. And there was a really well-funded general in the US military who was trying to do powered flight, but he was kind of doing these whistle-stop tours, lots of theatrics to it. Meanwhile, these two bicycle mechanics in Ohio, they had this idea for their flying machine. They built their prototype. They put it in boxes. They shipped it out to Kitty Hawk, and they went out there on a train. And they spent the summer out there. And they put it together, their machine. And they get in it, and they try to fly it. And it crashes. They fix it up and they do it again. And they keep getting it in.

And they do it all summer until either the machine or they

are too broken to go on. Because they're sitting in this death trap, right? Trying to fly it. And they pack it up. They go home. They take all their measurements, all their observations, and they go build the next version of their machine that winter. The next summer, they bring it back out. They fly it again. It crashes. They nearly die many times.

And eventually, after about I think it was six years of this, you have the famous historic first flight. And you think about this idea of experimentation being the prerequisite to innovation. And if the Wright Brothers could make this machine and go sit in it when every fiber of their being must have been saying, “Do not get back into this ferkakte machine!” Yet they did. Think about what we all can be doing with software, of rapidly experimenting and trying things and really building for our customers in a very iterative and experimental way.

That's why the more that you can do for developers to remove the friction, remove the barriers to running these experiments, whether it's cost, whether it's time, whether it's complexity, whether it's organizational decision making, then you actually create more experiments and it increases

your chances of actually innovating.

George Lee: That's great. Well, it's funny. The casual listener who drops in may think we we're doing a talk about aviation history and airline strategy, but, in fact, we're here to talk about your book, *Ask Your Developer,* which is a title I love. And unique in the sense that it probably started as a prescription drug warning, became a billboard, and then ended up in this expression as your book title.

Jeff Lawson: Yeah, well, you know, it's sort of funny. We had taken on a billboard a number of years ago. I think it was 2014 or 2015 in Silicon Valley. And we decided to take out a billboard because I actually think billboards are actually kind of useful in influencing decision makers. And as we were kind of moving up the org chart to more executive decision makers, they didn't know who Twilio was. And so we wanted to start getting our name out there. So, okay, we're going to get a billboard.

And we hired a firm, an advertising firm, to come up with their ideas for what our campaign should be. And they had just the worst ideas, right? So we said, “Okay, thank you.”

And we literally, we had this meeting. I remember it was on a Friday because the people doing the billboard were like, “We need the creative because it's supposed to go up on Monday. You've got to get it to us.” So we locked ourselves in a room and said, “Okay, what is this billboard going to say?”

And there were all these ideas getting batted around, but something that had been in the back of my head for years -- and you know those pharmaceutical commercials --

George Lee: Absolutely.

Jeff Lawson: -- where they, right? Where at the end, they say, “Ask your doctor if Regurgitan is right for you.” You know? For some reason, it was one of those things I would just think about in the shower. Like, “Ask your developer if Twilio is right for you.” And I just kind of blurted that out. I said, “What about 'ask your developer'?” And everyone in the room was like, “What does that mean?” I'm like, you know, I'm not even entirely sure, but we went with it.

And to me, it took us a while to kind of unpack what it meant because it kind of meant sense to us but we weren't

entirely sure why. But what it really meant was two things. First, on the surface level was, like, developers know what Twilio is. We now have 10 million developers on our platform. So the developers, the technical folks that you're relying on to go build the future of all of these products the companies are building in this digital era, in many ways, every company is building their future on their ability to go build amazing digital products. Well, those people actually know about Twilio. And so if you're wondering what Twilio is, you could literally ask your developer, and they would tell you. So that was at the surface level.

But then the one step below was this idea that, wait, it's not just like Twilio. It's actually developers, software developers are so close to the technologies that we all rely on, that we're building our futures based on, why aren't they actually our trusted partners in building these products and building our companies? Why are developers thought of as these digital assembly line workers? At most companies, executives don't really know that much about how software is made or who their developers are. They just kind of think of it as, “Oh, yeah, somebody writes a specification. Probably we have MBAs and they write a

specification. And the specification gets handed to developers to turn it into code." And it's almost like this assembly line where, like, oh, yeah, at the top, if we just put specs and Mountain Dew in, at the bottom side we get code and everyone's happy with that.

Meanwhile, at most companies, the salespeople are, like, "No." Like every executive, they know their top salespeople, their sales club, these are the rainmakers. And every executive knows their best salespeople by name and celebrates what they do. The same, however, could be said for developers in this digital world. But yet executives, business people don't really know what goes on in that technical world.

And so "ask your developer," really came to be a rallying call for shouldn't you really know what developers do all day? Shouldn't you really be leaning on them and asking them for their help and their partnership in building the future of the digital future of every company?

And then in a more literal sense, it also has come to mean how can companies actually leverage this talent? Because when I think about how most companies work, there's a

customer-facing group like product managers. Their job is to go understand customer requirements and write a specification doc. Hand the specification doc to those developers who then dutifully write the code, implementing it. And the way I think about that, that is telling a developer, “Here's the spec. Go write some code.” That's the “telling the developer.”

But I've seen amazing things happen when, instead of giving a developer a solution and saying, “Go implement it,” but instead of that, you give a developer a problem, and you ask them, “How can we solve this problem?” When you do that, you get an entirely different kind of engagement. You get different kinds of solutions. And you get software solutions that are built faster, that are built better, and solve the customer's problem better when you ask developers for their help in solving the problems as opposed to telling them, “Here's the solution. Go build.”

And that really is the thesis behind the book. Okay, great, how do you go do that? Because when you do that, it really creates a different outcome for your customers, for being able to attract and retain technical talent, and in terms of how you're going to go actually innovate in this

digital economy.

George Lee: There is this impression among business people that writing software is fundamentally a deterministic exercise. It's truly one of the most creative enterprises and activities in the world. So if you start to re-think that, there's a massive network effect. Like, the engineers get better at solving problems because they're closer to the problems, they feel more respected and enrolled. The business people begin to develop what you've called a software mindset. What does it mean for a businessperson, even though they may not have developed coding skills, to develop a software mindset that allows them to be successful in the world we're talking about?

Jeff Lawson: The software people are the people who wake up every day, and they see some problem in the world. And what their mind jumps to very quickly is: How can software help solve this problem? And that is not a skill set. It's a mindset. And it really leads you to this "ask your developer." If you're wondering how software can help solve this problem, ask your developer. But you have this gut instinct. "I bet if my computer system could put the right metric in front of those salespeople, we could solve

this problem. Now, let me ask my developer how we can go about doing that."

And so the software mindset is really one where you just see a problem, whether it's a business problem or global problem, a societal problem, a personal problem, whatever it is, and your instinct says, "I bet somehow if computers can actually get their hands on this data, then computers can help us solve this problem." And that's something that I see all over the place.

In fact, in company after company, you actually see this play out, especially in the world of hardware. Think about Stripe or Tesla actually. My favorite example, Tesla. You've probably all seen the car. It has almost no physical controls. It's just got a big screen. Why is that? Why is it just a big screen?

Well, the answer is because Elon Musk and Tesla broadly are software people, which is weird because you think of it as hardware. No, they're software people. Because once you're software people, what you realize is that you want the ability to continually morph and change and apply that agility of software at every problem you can imagine.

Because once you set something into plastic or steel, you can't change it. You can't software it later.

And so software people are always thinking about, like, "Hey, how do I preserve the future optionality to morph and change and better address my customer's problem as time goes on?" And that's why having everything in a screen that you can just do software updates for is really key as opposed to something that is fixed in plastic and steel that can never be changed again. And you see this playing out.

You see it playing out in so many different companies where what they're fundamentally doing is taking some domain of the world and pulling it into the world of software.

George Lee: You know, a lot of business people are trained to deliver a pretty fully-fledged product when they launch it into the market. And the idea that you talk about a lot in the book, the job's never done, agility is the key, experimentation is the key, iteration is the key. How have you watched big companies get comfortable with that?

Jeff Lawson: You know, some people take away this idea

of iteration or they'll have the first version, how do you deliver the first version? As the first version should be really bad. And that troubles a lot of business leaders. “Don't we have a reputation to upheld? Don't we have a brand name? What happens if people don't like it?” And there's a bunch of hacks, but you've probably heard the phrase “minimum viable product” as the idea of what's the smallest thing we can deliver that actually customers will value enough to consider a viable product?

At Twilio, we've actually changed it. We now call it a minimum lovable product. Just to try to make the sense of, hey, it's not just enough to be viable. You still want pride even in those first versions. But one of my favorite examples actually is the Apple Watch. Many of you may have an Apple Watch, right? And the thing that I find so interesting about the Apple Watch is that version one of it did essentially one thing well. It was a watch with very configurable face plates. Not face plates. The face, the design. Right? But it didn't do actually a whole lot. But it just said, “Hey, if you need a watch that's a lot more flexible than everything else and can talk to your phone.” And notifications actually are the main thing it did. That's all it did. So it did one thing, and it did it well. And it got

enough traction to buy them the ability to go on to version two.

And in version two they said, “Oh, actually, there's a heart rate monitor.” And then in version three there's -- I don't remember the exact detail but, like, oh, there's a phone built in. And then version four is, like, oh, and there's a blood monitor, whatever. And so every time, they add one thing that essentially makes the device better. I know they'll market 20 things, but really it's one thing that is enough value to a customer that enough of your customers say, “Yes, I want that.” And if you don't nail that one thing, well, then you've gone back to the drawing board.

And so what I encourage folks to think about is, in each iteration, have at least, no less than one thing that you really get right.

George Lee: This flips the thing on the head. I love the minimum lovable product, which is great. But again, traditional business people generally focus on a process which delivers a final product and then move on. The fact that something starts at minimum lovable and is advanceable from there is intimidating. In some ways, it

should be something you really embrace.

When I see business executives who really develop a software mindset, it's the idea that they can now take a product or service and make it better every single day ad infinitum and embrace that as a good, as an opportunity rather than something that's an obligation or a challenge.

Jeff Lawson: One of the ways that I like creating an environment where you can do that really well is, first, by focusing on having small teams.

George Lee: Yes.

Jeff Lawson: So small teams that are no more than ten people. The reason why I really like small teams is I think back to the early days of Twilio. We were a startup of ten people. On any given day, I might be doing customer service, I'd be on a sales call, I'd be ideating a new product, I'd be writing some code, and I'd be going to Costco to pick up snacks for the office. You do everything, right? And in that world, you can keep the whole universe of what's important and how these pieces fit together in your head.

But as companies get bigger and bigger and bigger, you start to divide up all these functions. And you've got the sales team and the support team. And you start to isolate those developers. In fact, it's all done with very good intention. You say, “We don't want to bother the developers doing customer support or hoping on sales calls.” So it's all well intentioned, but the net result is you end up isolating the people building those experiences and those products from the very customers they're trying to serve.

And I think this ends up getting very core to how do you actually deliver a minimum lovable product. How is a team going to know if their product is lovable, is minimum, is viable, is any of that if they don't actually know the customer? And know here's what we're trying to do, here's who the customer is? And so you've got to really keep those small teams A) with the full idea in their head of what they're trying to do and keep them in close proximity to the customer they're trying to serve.

And so I like defining teams by three things. Number one, a mission that they're here to serve. Number two, the customer they're here to serve. And number three, the

metrics of success. Because when we as leaders create a team -- and that's on us as leaders. We fund the team. We staff the team. But when you don't give them clarity about what they're here to do, then they can run around in circles and not know what they're here to do. So once you create a team and you establish here's the customer you're serving -- that can be internal or external. Here's the mission. What are you here to accomplish for that customer? And what are the agreed upon metrics of whether or not you're serving that mission successfully?

Once you establish those three, now that team can sprint every day to optimize and make those metrics better and better and better. And that's what I think is such a powerful thing because that unleashes the creative ability of that team because they know what's expected. They don't have to keep checking in and getting in alignment or agreement from everybody else. That is a very liberating idea to make that team able to go sprint for your customer every day.

George Lee: Without asking you to reveal any corporate secrets, maybe what are some of the bets, you know, one or two of the bets or experiments you're running that you're

excited about for Twilio in the years to come?

Jeff Lawson: Well, we have evolved Twilio from essentially a company with a couple of core products to one where we've got every mode of communication from voice, text, messaging, WhatsApp, email, chat, video, as well as now with segments, an acquisition we did last year, the leading customer data platform that helps you take data out of all the places where you see a customer, whether it's web apps, mobile apps, all this kind of stuff, and actually assemble one profile of that customer based on all these different streams of data. And huge volumes of data you're seeing about a customer, how do I actually use that to understand that customer?

And then we're taking all of that communications and all that data to go build our customer engagement platform. So we've got a contact center product. We've got a sales product, and we've got a marketing product. So that's flex, front line, and engage. And so really there's a lot of small experiments going on to create this whole picture. But what we know is that every company out there who's trying to execute in this digital era is struggling to go build this all themselves. Everybody needs to be as good as the digital

giants -- Amazon, Facebook, Google, Apple -- at understanding their customers, personalizing every interaction, making every interaction great so that you earn the hearts, minds, and wallets of your customers. And that's what we're really focused on building.

And there's so many experiments going on because every one of those is like a behavior or an adoption or a problem, a customer problem that we hypothesize that ladder up to how do we fulfill on this big vision that we have.

George Lee: I love it. That's funny. That vision speaks to many of the things you've talked about, which is something that started as a humble communications API, has become something that can fulfill that breadth and that compelling a vision. And so we could keep talking for hours, Jeff. Thank you so much, Jeffrey Lawson, CEO, co-founder of Twilio. Great discussion about his book, which I highly recommend, *Ask Your Developer.* Jeff, thank you for taking the time. And thanks to our audience as well.

Jeff Lawson: Thank you so much, George. And thanks.