

EPISODE 1158

[INTRODUCTION]

[00:00:00] JM: Staff engineer is a job title that suggests the engineer has deep expertise and considerable experience. More and more companies are adapting a staff engineer track, where an engineer can work to become a staff engineer. So what is the role of a staff engineer? Is it a management role or is it an individual contributor? What are the expectations and obligations of staff engineers?

Will Larson is an experienced engineer who has worked at Stripe and other prominent tech companies. He joins the show to talk about the role of staff engineering and the material that he is written about it, including an upcoming book.

[INTERVIEW]

[00:00:40] JM: Will, welcome back to the show.

[00:00:42] WL: Thank you so much for having me. It's, I guess – I don't know, like a year and a half or something like that. I'm excited to be back.

[00:00:47] JM: It has been a while. Simple question; what is a staff engineer?

[00:00:52] WL: You know, that's the right question. And it is funny, kind of classic management problem is you give people more advice. You come to realize you actually don't know the answer and you're kind of making it up on the fly. And so for me, a topic I realized I did not actually know the answer to and as I asked around, I came in to believe no one knows the answer to is the question you just asked me, which is what is a staff engineer? And that has been the motivation for the staffeng.com project that I've been working on and that I hope to turn into a book at some point in terms of interviewing 16, 17 folks taking what they said and turning it into kind of consolidated guides, synthesizing their ideas and some ideas of my own.

But if I were to try to answer that question really concretely, what I think a staff engineer is is to someone who was reached that senior engineer level, has that technical acumen and strength, but then moves into a leadership role. Not a management role, but a leadership role, leveraging their skills as a software engineer down this leadership track.

[00:01:54] JM: What does that mean in practice?

[00:01:56] WL: Yes, that's a great question, and that's where I've developed this idea. And it's not a new idea. A bunch of different companies have done this before. Facebook has their own set, but really tried to identify a group of archetypes in terms of like what to the engineers actually do. At some companies you see folks who are this architect, and they are potentially not even writing software anymore. They are only designing systems, working on the big boxes, making tradeoff decisions, coordinating across teams.

Other companies you see – And this is probably the most common version of the staff engineer. So you have this tech lead, who is the senior most member of a team who is setting a lot of the technical direction for a specific team, but is really still in the weeds, who is programming, who is partnering across teams to figure out like the actual goals of one particular team.

Another one that I saw, the rarest variety by a longshot, but I see it in a handful of folks that I spoke to from Michelle Bu at Stripe, to Rick Boone at Uber, which I call this right-hand role. But basically this is a senior individual contributor peer to a vice president, to a chief technology officer or chief product officer or something like that who basically operates the full authority of that CPO, VP of engineering, but only within a narrow subsection. For example, API design. For example, SRE practices or something like that. And there's at least one other archetype that I found as well. And so what I think is really important and one of the reasons why it's so challenging to answer the question of like what genuinely is a staff engineer is that different companies take a different view on what they actually want that person to do. And many companies end up having like all four archetypes represented over time, which makes it even more confusing to talk about what does a staff engineer genuinely do.

[00:03:49] JM: Can you say a bit more about these archetypes?

[00:03:51] WL: Yeah. So I went into this project with a really clear sense of what I wanted a staff engineer to be, but I think it's important as someone trying to actually shift the industry a little bit. And genuinely, that's my goal with this project, is how can I find a topic where I can actually push the industry in a meaningful way where having completed a year, 18 months of work on the topic, that the industry has changed?

I was thinking, "How can I shift the industry towards the version of what I think a staff engineer should be who is collaborative, who partners, who listens, who is technically excellent, but is a leader beyond their technical strength and is greater within the company and more impactful because of it?" And so I was hoping I'd find kind of a number of examples of this perfect embodiment of a leader that I've thought about. And what I found is that there are just so many different folks with different styles, and that if I tried to simmer it down, just this perspective I came in with, it didn't make any sense. And that's where I found four archetypes that I've found valuable, this architects, this right-hand, this tech lead. And then one may think is a little bit convoluted, I call them the solver, where you often find his fixers.

There is an old blog post. It might've been from Rands about these free electrons, people who just kind of bounce around and solve problems in organizations. And really the solver is another version of this idea, the free electron, who just is directed by leadership to solve some sort of really important problem that isn't necessarily shaped in the right way for a team to take it on or might not be the sort of thing you want to do a durable investment in. But I think this is helpful, because so often we'll get this question from someone about like, "Hey, I really want to be a staff engineer." And part of this is we've really created this inevitability of progress in career ladders where folks just assume they'll kind of step on and ride an escalator somewhere.

But the staff engineer is not really like that. It tends to be that the first non-career level at a company. So this career level is this idea of a place where everyone reaches eventually. And if you don't reach eventually, then you kind of don't succeed at that particular company. And so typically it's a senior engineer. And if you don't reach senior engineer in, say, seven years at a company, or maybe eight years at a company, then probably something is like not quite working for you and that company together.

But staff engineer is not like that. It's not just a stay in place and do what you're asked role is. It's a genuine leadership role, and that means that you can't follow a path. It means that finding the map yourself is part of what makes you that level of leader, but it is hopefully the people a little bit of a map so they're not just feeling blindly. And particularly, they're not chasing dead-ends. And the most common dead-end that people really want to follow is like how do I become a staff engineer who doesn't deal with coordination and only does technical work? And the point of view in my work is that that role doesn't exist. There is no purely technical staff engineer role. It's just not leadership when you operate in a purely technical fashion.

And I just haven't found any examples of someone who does in a successful company. I'm sure there are some one-off examples out there. So don't have to go research to find the only exceptions, but I would argue that's actually just a misapplication of the title, not a counter evidence that that is actually how the role works. But really want to get people these maps by having different archetypes where people can kind of see which of the versions do I want to become. I think we give people much better advice about how to pursue that, while also recognizing for leadership roles. You don't simply follow the map. You have to take the map. You have to think about it. You have to understand the context you're in and then figure out the path a little bit yourself.

[00:07:52] JM: So it sounds like in almost every case, the staff engineer requires some amount of technical expertise/coding and some managerial expertise, or people skills.

[00:08:04] WL: So definitely the first one is true, and every staff engineer I've spoken with is someone who was like an excellent senior engineer at some point in their career who was just excellent in the technical work. On the management side, what are management skills? And I think it's a really common mistake to combine this idea of management and leadership into a single idea. And I do want to tease them apart. So some things that are management skills are career management, giving advice to folks on their career. Giving good performance reviews, like systematically running sprints or road-mapping or sort of the mechanics of actually running an organization. When I think about management skills, running a hiring process where you get kind of great folks in and have a good candidate experience for the folks who come through it.

These are all like org design, like how should we structure these teams? When should we do a reorganization? How do we do reorgs in an effective way? These are all management skills, but when I think about leadership skills, I think they go a little bit more general, like critical thinking, judgment, listening and understanding, communicating, empathy a core leadership skill. And so I don't want to say that you have to have management skills to be a staff engineer. You don't need to be someone who's capable of running like a well-thought out or designed organizational restructuring, but you do have to have these core leadership skills, which are kind of, in many ways, like these fundamentally liberal arts skills.

And just as one anecdote that I think is kind of interesting. As I talk to folks, some of them have been managers in their path to staff engineer, and the majority of them have not been managers and don't want to be managers. But I have found that the folks who do a detour into management really do benefit from building out that awareness, building out, I think, some of the leadership skills, which are easier to practice when you're in this managerial role.

And so I do think if you're considering management and you're the sort of person who would go into management for the right reasons. It's like caring about people, wanting to help, wanting to execute, wanting to have a huge impact both on the team but also on your company. Doing a detour into management and does generally help people become more effective at staff engineers later in their career.

[00:10:28] JM: So let's say I am an engineer at a company and I the vision for wanting to be a staff engineer. Are there concrete steps that I can or should be taking?

[00:10:38] WL: Yeah. So there definitely are, and the exact concrete steps that you should take are going to depend a little bit on the size and style of company that you're at. If you're in a really small company, say, less than 50 people or less than 100 people, then these are typically companies that don't have much structure or much process, and you can't just work the system correctly and kind of get to the place that you want to.

In a really small company, the things that are going to matter the most are having disproportionate impacts on things that the company values. Communicating clearly with your manager that this is something you're focused on and matters to you, and just making sure they

actually value you individually. And some of these are, I think, hard to control. But there's this idea of managing up, which a lot of people think shouldn't matter. But a great example of managing up in the positive is that if you're excited and high-energy, and people want to work with you, even if your work quality is not higher than someone who folks find to be draining to be around, it's like you're significantly more likely to get promoted into a staff role. And so just the fact that folks feel good about you and want to work with you that you're an energizer, rather than someone who detracts from the energy or even just neutral really matters. And I think it particularly matters in a smaller company when these things tend to be more field-based. But really want to emphasize this. If your manager doesn't know that you want to move into a staff role, you are extremely unlikely to move into a staff role. So communicating that is really important.

And this other idea that a lot of times engineers get caught up on this idea of the thing that the company should value and putting all their energy into the thing that the company opt to care about. So this could be performance. This could be scalability. This could be certain types of security. And so it's really hard to get moved up into the next level when you do work that you think the company should care about, but you keep getting signals that it doesn't.

A huge part of being in a leadership role is aligning with other leaders. And if you show your inability to align with other leaders by continually trying to convince people that value things that they genuinely don't, you're in fact demonstrating that you're not hitting that leadership level yet. And this can be disappointing, because there are things that company should care about that they don't, and there are I think huge numbers of examples of that. But I think it's also part of moving to this leadership role is recognizing that your perspective is just one perspective, and that the folks who are making other decisions have like really important and valuable perspective as well around you.

Stepping back from the small case though, if you think about like a larger company, larger companies typically have like a pretty structured process where there's a promotion process, say, once a year or twice a year. There might be a calibration process where these performance packets are discussed across the entire management team, or if you're larger, just like the engineering team, or if you're really large, just a division within the engineering organization.

And then from that, every company says they don't do occurred anymore in terms of this many people get promoted and this man can't. But fundamentally, there's always a budget somewhere determining kind of how many folks can get promoted, determining how many folks can get raises. And every attempt to pretend there isn't a budget constraint somewhere is in fact a smokescreen. There is always a budget somewhere that's controlling a lot of the outcomes here.

But within these like more structured kind of larger company processes, so step one is still make sure your manager knows what you value. Step two on this idea, this promotion packet, and Julia Evans has a great blog post about drag documents, which is really the same idea, but writing down what you've done and why it matters. And then the thing that I think is important is not just doing that in isolation. A lot of folks don't recognize their own best work or their own best impacts. So take that promotion packet that you write yourself, review it with your peers, like, "Hey, what's missing? What is undersold here? What stuff that actually didn't go well that I should stop representing as a win?" Review that with your manager on a recurring basis so that they're armed with awareness about your work. So that's a little bit less work for them to repair this successful packet for you. And really getting that manager aligned with you that you are doing staff work and should be in that role I think can't be overstated to how important that is.

Moving into a leader role is a company taking a bet on you as a leader, and they don't do that on folks who they're uncertain about. And this again to this point about like the escalators stops at this career level. Anything beyond that, you really have to do deliberate work to kind of carve the path yourself. The last thing I'd say on this topic is that there is this pervasive idea of the staff project where you're a senior engineer. You do something of an inordinate complexity, that really matched the company and that justifies her promotion to the staff level. And a number of companies do have this.

When I was talking to [inaudible 00:15:58] at Dropbox, like Dropbox did have this idea of a staff project. And others were familiar with the idea, but almost no one seems to actually do these staff projects. As I've done more of these interviews, effectively only one person had a clear example of doing a staff project. So that's one kind of meme or idea that I think is just genuinely false and worth kind of reconsidering if that's been what you've been telling folks, "Is this a hard requirement to reach the level?"

[00:16:28] JM: Is there a difference between the terms staff engineer and tech lead?

[00:16:34] WL: This is a really good question. The answer is like this is a place where the industry doesn't have a ton of standardization. So at Redfin, which is where my wife is a software engineer, they have this idea of a tech lead, which is this level beyond senior. So it's kind of mid-level engineer, a senior engineer, a tech lead. And I think they might be principal or architect after that. I forget the next step for them.

I was talking to someone at Stitch Fix recently working on a story with them, and the same thing, so senior tech lead principal as well. So they're kind of like a little bit of an exception though. Most companies that I've spoken to, tech lead is more of a role than a title where you might be the tech lead or the team lead on a specific team, which is like your role in a given project or your role on a given team, but it doesn't actually necessarily mean you're paid differently or are expected to perform at a different level than another senior engineer. It just means you're doing different specific expectations in your work right now.

The most common title kind of hierarchy is junior engineer, early code engineer, software engineer, senior staff, senior staff, principal, senior principal, and vendor. There's really no consistency beyond that, but distinguish is one that comes up, or architect is one that comes up. Very few companies have all of these roles for what it's worth. Most companies might have senior, staff senior staff, or senior staff principal. It's really a bit of a bedlam out there, and this is why there're sites like levels at FYI, which are trying to – Or it might be progressions.fry, which are trying to figure out like how do these different levels and titles map across different companies, that there isn't a ton of consistency, and tech lead can be a staff at a number of companies, but can also mean absolutely nothing. And that's where I think you have to drill in a little bit into the actual style of work. And that's why in the Staffeng project itself, I looked at tech lead as a type of staff engineer as supposed to a level or a title specifically.

[00:18:47] JM: As you talk to this variety of staff engineers and tech leads and the different people that you've interviewed for your website and for your book, what are some of the commonalities that have come out? What are the common lessons that have shown up time and time again in your conversations?

[00:19:02] WL: Yeah, there are a lot of lessons. So just taking a handful of them. So one think it's that reaching these levels is a deliberate outcome. It's pretty rare that this just happens. You have to be intentional about making this happen for you. Two, another lesson that I think is important is that the title does matter for the folks who get it. It particularly seems to matter for folks who are black, for folks who are women where they – And certainly, kind of inner sexuality, like the one black woman I've spoken to in the project so far. Having the title really changed how people treated them. And it reduced the friction and just there every day work in a really important way.

And so, conversely, some of the other folks I spoke to didn't feel like the title change things for them that much. So this is where I think the titles can be particularly impactful in terms of clearing the road for folks who do work that your organization might not have the right instincts around recognizing are already senior leaders in your midst. So that's one of the things that I think is super important.

Being visible – I think a lot of the staff engineers talked about the work they do to show their work and their team's work. The goal for any organization is that it should be able to evaluate people's work fairly and consistently without you having to do any kind of PR for your work internally, but effectively no organizations work that way. And the further you get from the core of what the company is doing, we're kind of this – The work is like obviously aligned with what your CEO of your senior leadership team cares about. The more time to spend in terms of just like showing people why the work you do matters. Being an effective storyteller is a huge part of being a successful staff engineer at larger companies in terms of like showing why your work is so impactful.

This idea of alignment I think is really, really important as well. There's this idea that once I get some title, once I become a staff engineer, or once I become a principal engineer, I'm finally going to get to do what I want. I'm finally going to get to tell people what I want them to do. But the hidden reality is that these are leadership roles, which means that you're held to like a very high standard of alignment with the more senior leadership of the company where you do have more authority, but you're only allowed to use the authority in complete alignment with the more senior leadership at your company, where you can only do the things that they are aligned with

you doing. You can't actually do the things you want to do that don't align with them or you will fail out of the leadership role pretty, pretty quickly. So another thing that I think folks realize is that the magnitude of your vector increases as you move into these leadership roles, but the actual directions you're allowed to point it tends to go down.

Another one is coaching and mentoring is a huge part of this role. Again, not a pure technical role where you are alone in a room and someone is clearing the route for you. It's more you're the one who's clearing the route for others is a really important piece.

Those are some of the pieces that I think are really important, but there's also a lot of things – Yeah, okay. Maybe the last one I point out, which actually I should have brought up a little bit earlier is building your network externally. There isn't a playbook out there about how to be a great staff engineer, and I hope this resource helps with that. But the place that people go to for help as staff engineers almost exclusively is other peers in similar roles at other companies and just learning within the network.

So much of the work you're doing is a spoke or really particular or context-dependent where there is never going to be a book that can give you like the full strategy to solve your problem. Books can give you hints, ideas, patterns, but it's really this peer group that you develop over years who you trust, who trust you and give you like unfiltered advice and feedback as well. In these leadership roles, people stop telling you when your ideas are terrible, and sometimes they'll just go along with them as you fail kind of quite publicly and without much feedback as to why you're not doing a more successful job. So yeah, building your network is another theme that came up a number of times.

[00:23:18] JM: Why is the subject of staff engineering interesting enough to write an entire book about?

[00:23:24] WL: Yeah, that's a great question. And I think it comes down to what makes it worth writing a book? What's the point of writing a book? And then also kind of the reaction to like what did I take away from my first book, *An Elegant Puzzle*, that I wanted to do differently for the second book?

So maybe starting with the third question. So I wrote *An Elegant Puzzle*. Kind of half of it I wrote over 10 years in blog posts, and the other half I wrote in a four-month sprint to kind of finish out the book. But as you read it, it's literally disjointed, right? And this is what happens when it wasn't designed as a book. It was designed as a series of essays that happen to have collated into a book along with a lot of essays written trying to join the pieces together. But it was a little bit of Frankensteinian in its construction, and it wasn't super pristine.

So when I thought about writing the next book, I wanted to find something that I could do incrementally, but with more intentional about what I was trying to accomplish. And I was really trying to think about where is a place I could write something that was new that wasn't missing? Again, I spoke with this earlier, that I thought could push the industry forward in some sort of meaningful way.

And so as you look in the management kind of book domain, there's a ton of great books out there. Camille's book is phenomenal, *Making of a Manager* is phenomenal. There are so many great books out there that I don't think I think I have a whole lot more to say about management right now that hasn't been said by a number of really talented voices.

Conversely, as you think about senior engineer books or for folks – Really, folks who are trying to move past senior engineer roles in like this staffeng+ roles, almost nothing has been written, but a lot of the works are very similar to management work, but it's just this like leadership work. And as I had more conversations with folks trying to make this leap, and also as I tried to do organizational design in my own roles where I thought about like, “How do I actually involve these staff engineers successfully in org design?” Because not many companies out there have great patterns about how you make staff engineers like leaders with access to information as supposed to just like more senior, senior engineers.

And so I just really felt like this is a project I could complete, a project that could be genuinely novel and valuable to the folks who consumed it. Project where people had an appetite for it, and one where it could be done in a relatively bounded amount of time. It's not a project that I think will benefit from exhaustive research, but I think roughly 20 interviews, roughly 20 kind of guides compiled from these interviews, and it could be a useful thing to get out.

And so those are basically the reasons I decided I want to work on this. And I feel like I'm 80% done, and that's the sort of lie you tell yourself repeatedly over months to actually get to 100% at some point. Maybe I'm more like 50% done, or 80% done with the first pass. And then there are a lot of passes to go in terms of improving the quality, the editing, the sequencing and so on and so forth. But it just felt like a project that could actually help people. And so to me, that was a pretty exciting idea.

[00:26:37] JM: This is kind of a boring topic, but salary compensation I think can be complex and negotiate for with a staff engineer role, because you don't necessarily know where you're negotiating from. There's very often like just a few staff engineers at the company. It's hard to find out what they're getting paid. Any advice for how to approach salary negotiation for a prospective staff engineer?

[00:27:01] WL: Yeah, this is a hard question. I think a couple things happen, which is that some companies treat a staff engineer effectively like they would a senior engineer, but with like a little bit of a premium from the compensation side. And some will treat a staff engineer as like a genuine leadership role where there is a lot of flexibility around the comp bands. There's a lot of flexibility around everything. And so it's pretty hard to figure out which of those you are actually in. And so rule of thumb is like small companies – Really, small companies, like companies like 10 to 20 people who don't have any structure or system and they're sort of making up every offer in a bespoke way. So that's one world where things are like quite open-ended for everyone in such a small company.

Honestly, I'd be really skeptical of a 10% or 20% company offering you a staff engineer role. What does that mean, really? Can you do staff level work at most 10 to 20-person companies? And I would argue that it's really hard to tell at a 10 or 20-person company with five engineers whether someone is or isn't doing staff level work. Certainly it doesn't mean they are not capable of it. It's just so much the work at that level is really rapid execution and not necessarily high-quality execution that thinking about it, really hard to tell.

Then there's kind of like imagine from, say, companies with more than 20 people and less than like 300 or 400, and these companies are probably treating you just like a very senior, senior engineer or something like that. Maybe a little bit more comp. Well, probably a little bit more

comp, but they're not going to do much bespoke for you. But at some point when you're interviewing with a company with a couple thousand engineers where you're, say, one of the top 1% of the engineers by level. Things do open up for you.

A real story from earlier this year is I was talking with someone interviewing for a senior staff role, and they were trying to figure out their offer, and they talked to a friend of theirs who I interviewed for an engineering director level role at the same company and they figured out that their friend's comp, and then effectively got the company to almost match it. It was like modified in the details, but very close. And so this person for a senior staff role got an eng director level offer. And then after the director level, there were more comps. Well, their friend had a comp, but in general on glass door, or blind, or whatever. There's going to be like more comps on the management side typically, because there are, at most companies, more directors than staff engineers at a certain point in their development. Although as a company matures, probably the inverse would be true at some point.

So I think basing on the management side of the latter can be helpful and is really the best bet in terms of getting fair compensation for the role at a certain level of company. But even managers at smaller companies with less than, say, like hundred engineers, things are just like much more rigid typically at that level. So I wouldn't try to use any like savvy techniques for smaller companies for the most part.

But yeah, I think you just have to be willing to walk away, as is like typically the case for these interviews, that if you bring something really unique and special for the company that you believe will meaningfully change their outcomes, they're going to be willing to make things shift for you a little bit in a way where if you're just coming into the process kind of operating it in a pretty standard way, you won't.

The last thing I'd share is as I personally did at mind my last job hunt, I spoke with Julia Grace, now at Apple. At the time, at Slack, and she gave me a great piece of advice, which is basically don't play team sports alone, which I think she got from a coach of hers. But this idea that we've been told essentially that this interview process is a private thing. She kind of follow the lines that are painted on the ground. But in leadership roles, like the vast majority of folks interviewing are not following the lines painted on the ground. They have back door references into the

company. They are talking to the founders or the executive team in building a relationship there and really not trying to just do what you think is normal based on what works for you as a senior engineer or earlier roles, which are more commoditized from a hiring perspective. They're really thinking about treating it as a bespoke artisanal role in approaching it from that perspective in terms of getting the offer that would be the best for you.

[00:31:33] JM: What's the difference in impact that a staff engineer should expect to have versus a senior level engineer? Is there something more impactful? Or is it just a title?

[00:31:43] WL: Measuring impacted is always really challenging, but I can give some examples of kind of the minimum of what I'd expect from a staff engineer. So I think there's always this tradeoff when you run into a challenging problem between like just doing something that improves it or finding like a system-level solution. And I want a staff engineer to have great judgment around doing both of those and doing those both really effectively.

So a good example is Calm, where I am the CTO. We were having some latency in reviewing poll requests, and one of the staff engineers just basically paused, like, "Hey, we should just like automatically ping people over Slack when this happens." And then he just went and worked with one of the numbers of our DevOps team and they just like have gotten that done. A different person could like make that suggestion and then kind of sit on it. But a staff engineer I would expect, when they have that idea, if it's something that's like relatively straightforward to do, just go and does it. And I think that's incredibly important.

Another example of one of our staff engineers at Calm is that she joined our partnerships B2B engineering team. Our approach at integrating these different partners was a little bit frictionfull from the engineering side, and she just looked at them, found the pattern, and build something that allows us to onboard them much more quickly, where I think a senior engineer might focus on like getting them on-boarded individually as quickly as possible, or kind of focusing on the detail, or they might point out the problem, but not self-authorize to go fix it themselves.

And so this idea of executive function I think is a huge part of it, where not just pointing out problems, but genuinely fixing things, like this bias for fixing and bias for change and bias for

impact, not just bias for observation. There are more problems that any company can be fixed. So pointing out problems at some point is not super helpful.

But also the staff engineer, I think it's like having the ability to like see yourself and notice when you're just spinning or burning yourself out doing these like little fixes and then stepping back and like advocating for the more systemic change, which might in this case be, for example, spinning up a tooling team internally who's able to actually work fulltime on these sorts of improvements or something like that.

So to me, it's really self-authorizing to make the small improvements first, and trying that. Not just making observations. Also not leaping to the systemic fix. Rolling up processes, reeducating the team about how we want to do different types of work is really heavy and really expensive. When necessary, you need a staff engineer who can go convince the team to do a new style of work or a new approach design in a new way. But you don't want people who jump to that before they've done the quick easy ones that might be enough. So you can go back to focusing on actually implementing the features and functionality that your users actually want. Your users don't care how you do sprints. They just care if you're improving the product on like a regular basis. So that's sort of the different style of work.

In terms of impacts, measuring engineering impact is pretty disastrous still. From the book *Accelerate*, we have this kind of four core metrics for understanding developer velocity, but it's really more about an organization in terms of understanding how frequently are deploys failing? How quickly can you revert? How long does it take from a change getting open, to getting fully deployed? How big are changes that get deployed? Those are useful to understand, like an organization's health. But they aren't pretty helpful for understanding an individual's impacts.

At Calm, the way we've been thinking about impact for engineering is kind of on three dimensions. Just the number of kind of experiments or AB test we run each quarter, and we have like a fixed targeted amount that we try to do every quarter to make sure we don't need to experiment. And that number is based on just our traffic volume and like getting this statistical significance on those tests. Two, the number of kind of product features that we want to ship, and we kind of try to break this into like incremental improvements and kind of bigger bets. And

then also just the number of incidents that we have. And this last one is not like a perfect goal as we develop a more in-depth incident program. There are lots of things we do to improve that.

What we really care is the number of like user impacting incidents in driving that down. Not driving down the total number of incidents, which is sort of like the wrong thing to think about. But even that, it's much more about organizational function, rather than individual impact. And so to me, when I think about identifying a staff engineer, it's more looking at the behaviors and the practices and the approach they take and looking at the overall organizational's health within engineering. Kind of doing a bit of an intersection there, measuring the impact of individuals. I think at a high-level, as supposed in the details of their really specific work I think is quite a challenge.

[00:36:40] JM: Were you a staff engineer when you're at Stripe?

[00:36:43] WL: Yeah, that's a great question. So the last like 3 to 6 months I was there, I was a Stripe tech lead, but I think that was sort of an experimental role, which I think I didn't really end up transitioning into. In practice, I was still doing my kind of managerial role leading the foundation organization at the time that I decided to kind of transition out to Calm. So it wasn't – It's sort of from a purely title-based perspective, that's kind of true. But I think it would be somewhat disingenuous to say that I actually was.

[00:37:15] JM: Well, what was your personal experience in being a pseudo-staff engineer?

[00:37:20] WL: So one, I think it was pretty hard to actually make the full cutover. I think in practice, my focus there was kind of backfilling my previous role in terms of finding someone else to step into it. So wasn't able to do as much as I really hope to be able to do from just like an attention perspective. But one of the ideas that I have, and I wrote an article about it a couple years ago, which is kind of *A 40-year Career* is the title of it, is thinking about I think to have a sustainable career. And a lot of times in Silicon Valley, in technology, we don't think about sustainable career as we think about, "Oh! Like this American dream. I'll go into the jungle and come out at 21 a rich man." And so sustainability is irrelevant for us, right? But I really think that's the idea of like what if we're trying to work for solid 40 years in this technology profession?

I think to do that, you have to have periods where you switch between really broad focus and kind of getting like narrower, more deeper focus.

And so to me, the intention was really how do I deepen and narrow my focus for a while and kind of rebuild the reservoir of deep work that I've built over time where management, particularly managing a like larger organizations or companies, vary interrupt-driven. Huge amount of scope, but tends to be relatively narrow on things. You only go as deep as you need to to solve, and then you move on to the next thing. You rarely go deeper than minimally necessary. And I think that work is really motivating and from an impact perspective, but a mix I think is important for me anyway.

And so I got to really focus on reliability, multi-region strategy and then our cost, like our infrastructure cost strategy. But again, that was really like partnering with these like amazing teams that we have in place there. Smruti Patel on the efficiency side there. Just like amazing engineering manager and a phenomenal team. [inaudible 00:39:16] and a number of other folks, Grace on the reliability side. So just really phenomenal folks. And so for me, it was more risk getting to come in and help shape the approach and thinking a little bit on it. I just didn't get much of a chance to go as far or as deep as I would have if I had decided to stay there a little bit longer. But conceptually, I think the role really makes sense.

And one of the interviews I did was with Michelle Lou, who has done something similar open on the payment side, and I think she's just done like an utterly phenomenal job of taking this really important series of constraints in terms of kind of particularly like API design and interface design, but not exclusively that for payments. And payments is just this massively complicated footprint underneath and how do you actually make it accessible for your users who don't really care about payments per se? They just want to accept payments and move on understanding the full complexity. So, I think, structurally, the role was well-designed and really thoughtful and just like didn't –The detail is in timing. It ended up like not quite seeing it come to fruition for me, personally.

[00:40:26] JM: As we begin to wind down, is there anything else you want to add about staff engineering or your process of writing the book?

[00:40:32] WL: To me, like the core thing – I said this a few times. I just want to like kind of throw it back out there, is that staff engineering is genuinely a leadership role and it's so important to think about it from that perspective if you want to find one of these titles and be successful in it.

Another thing that we didn't talk about that I think is really important is that some people just have to switch companies to get that role. And I think that's not necessarily a reflection on you. I think folks often stay in situations where their skills are not as valuable for really long periods of time. But as I've talked to more folks in this interview, it's like many of them find that they switch roles or switch companies and get the title relatively easily where they had been toiling for years to actually get into that role.

So if you're doing all the right things and it's not working, I think it's time to do a little bit of a checkup. Are you in the right situation for you? And if you're not, like I think don't be too afraid to go talk to some other companies. I bet you can get the title at another one of them. Another idea though, and this came from an interview with Michelle, is that you should be really deliberate about testing that you want this role. It's a totally different job. It's not necessarily the job you think it is. And each company has a different version of it. But I think this kind of perception that you should be on this career escalator to get to the next rung really does trick people into taking on jobs they, in retrospect, regret taking. So just find a way to test parts of the role. Make sure you actually want it. If you just don't want the job, don't take it. It just helps to be a little bit deliberate on that one. If it's just a status symbol or if you think if I just had the title, then I promise you, just the title, it's unlikely to solve your core problem.

And then just in terms of writing the book. Man! I think it's been a lot of fun to work on this one, and I'm really hopeful to get it into something early next year that I can kind of ship and go out and see what folks think about it. I think there are all these little topics that are hard for many publishers to kind of write to. A few folks that I've spoken to, Tonya Riley, in particular, have talked to some publishers about writing on this topic, and Tonya Riley is like the number one person you'd want to see write on this topic. She's just done phenomenal work at Squarespace, and before that, at Google. But a lot of traditional publishers don't feel like this is a large enough niche to actually write words. Like they won't sell enough books for it to be financially successful, and I think this is like one of the powers of kind of the independent publishing or

alternative publishers like Stripe, which have like a different kind of set of incentives out there, Stripe Press that is, incentives that they're operating against.

And so really hopeful to get this out and hopefully it sells enough copies that it creates space for more people to write about the topic, rather than being the final word on it. I think there are so many other people out there who have better versions of this book to write, and I look forward to reading those overtime as well.

[00:43:43] JM: Will, thanks for coming on the show. It's been great talking.

[00:43:45] WL: Yeah. Thank you so much for having me again.

[END]