

[JAVI BELTRAN, "TELEBOT" PLAYING]

SPEAKER: Welcome to season seven of *The Prodcast*, Google's podcast about site reliability engineering and production software. This season, we are exploring what it takes to ship software, and what that even means to different folks. Join us as we discuss how to ship it as SREs. And of course, the most important part is the friends we made along the way. Happy listening, and may all your incidents be novel.

STEVE MCGHEE: Hey, everyone. Welcome back to *The Prodcast*. This is Google's podcast on SRE and production software. I'm Steve McGhee again. It's me. You've heard of my voice before I think. I also I have my friend Matt here. Hi, Matt. How are you doing?

MATT SIEGLER: Hey there, Steve. Matt Siegler. Welcome back. I'm happy to be back too.

STEVE MCGHEE: It's funny because we always talk about temporal things about, it's been a while and stuff. And then people listen to these two years later and time doesn't matter. But that's fine.

MATT SIEGLER: Oh, yeah. And they listen to them sequentially, like, in long binges. So that doesn't make any sense to them either.

STEVE MCGHEE: Yeah. So hello, whoever you are, sitting in an airport somewhere listening to a series of 15 podcasts that you've had downloaded to your device. You're the one that we're talking to today. Speaking of talking to today-- see how I did that?-- we have our friend, Daniel Hobe, live from San Francisco, California, I believe. Daniel, who are you? And why are you here? And how's it going?

DANIEL HOBE: Yeah, hi. Great to be here. I am Daniel Hobe. I've been at Google for 23 years, so a while, working on a bunch of things. Have been an SRE since 2007, I think. But joined Google in the sort of corporate IT world working on our email systems and filers before Gmail and all of that stuff. I've really focused most of my time on consumer products. So Search was the first thing I really worked on.

And these days, I run the SRE teams for Search, for Geo, GDM, Google DeepMind, and Labs. So kind of a fair amount there and a lot of different touch points on the different products that we've got. Yeah, but like I said, really happy to be here.

STEVE MCGHEE: Cool. We never really worked together, but I feel like we worked adjacent to each other. We were in the same office for a little while. I was always in these other non-Searchy bits part of Google. And now I'm in Cloud. So I'm like, you have no presence in my world. I'm sure we overlap a little bit in some places, but it is kind of funny because we've been around a similar amount of time.

DANIEL HOBE: Yeah.

STEVE MCGHEE: But Matt though, you are in that world, aren't you?

MATT SIEGLER: Yeah, and I kind of feel like I've floated through the org chart dodging you in a few different ways because I went through Shopping and through the various positions in the organization. And now I'm effectively within Gemini's relationship as an SRE. So we are now actually interconnected, which is wonderful.

STEVE MCGHEE: Nice. It's a big company, but it's a small org? I don't know. Something like that.

So the thing that I think we'd like to talk to you about, our theme of this season is shipping, or ship it, or to be shipped, or something like that. And your products that you support handle a lot of people, it turns out. Like, this whole thing, Google Search, has been around for a while.

People have heard of it. I think some of our users may be using it right now.

So we're talking about billions of users. And so what are the challenges around shipping to very large things? I imagine that not everything has still been solved to this day.

DANIEL HOBE: No, I think it's a unique challenge. And I think one of the things that's kind of kept me here has been like, I don't think there's many other products in the world that have some of the same challenges that we do.

And so we talk about Google Search, but it's Search but on many different surfaces. And there's the Search app, and then there's the web interface, and then there's showing up in many different features within the app. So it's a hugely multi-faceted product, similar to Geo is also hugely impactful a lot of users.

I mean, I think the challenge is kind of what I alluded to before just now, is all of the different features. And you think about Search has been around for 25 years, I think, or more at this point. And you think about all of the features that have been added, and eras of technology that that product has been through, from a desktop-focused early internet kind of product to the mobile revolution.

STEVE MCGHEE: Mobile first, baby. That's right.

DANIEL HOBE: Yeah, social, and now we've got the AI focus. And all of those things, we've developed and changed the product, and added features over those eras. And they all need to keep working. And so how do you know your product is working when it's that breadth of features? And that's, I think, one of the largest challenges.

And I think as we talk about this, and I'm sure this will resonate with a lot of folks, is typically the way that you would do this, to make sure you're doing it right, you're getting the right thing out the door, is it takes time. So you kind of are moving a little bit more slowly. And now, as we're trying to iterate quickly on the product, you have a lot of pressure to get things out to users, to get feedback much more quickly than we might have been used to.

And so trying to find the right balance through all of these eras, there's never a right answer, as I think you're kind of saying. So that's one of the biggest challenges that I think we've got. And you think about all of the various dimensions that you're working through. So just Search. Just typing into the Search box. Pretty straightforward.

But now, you have personalization, because we know you a little better. We probably know what you actually wanted to see. We can help with that. That's an additional sort of dimension. Now you think about what device you're on, that's another dimension. What language are you in? What country are you in? Just at that macro level, you start to get this explosion of different dimensions that you need to understand, is this thing working? And so I think that's really a continual process that we try and address.

STEVE MCGHEE: So what I'm hearing from you is that it's not one size fits all, for sure. There's many ways to do this. And it probably, I assume, has changed over time. And I want to get to the new and hip and modern things that we're doing now. But I'm curious, what if you had to briefly speedrun the last 20 years of release processes real quick for our audience? I'm sure that wouldn't be very difficult. But my real question is, we talk about it in the book and stuff. Like, we've published some stuff about, you can do canaries, and you can do experiments, and stuff like that.

How'd those go? Are those really the things? Or is there more secret sauce to be revealed than that? For someone who's just getting going, you know what I mean?

MATT SIEGLER: Yeah, I heard we built a lot of stuff to make this safe to do. And we're really good at it. So but before, no one had it internally, and then we built it, and now we do it. And that's just how stuff goes, and now we have SRE, and we launch things. So I think speedrun is the right model. So what happens? How do we do stuff? How do we ship?

DANIEL HOBE: Because it's slightly different for each product, but sort of to pick on Search and generalize a little bit. It really is kind of what we describe. There's canary gating, there's experiments, there's slow rollouts and step function as we get to a larger audience. And then

A/B tests, what are we seeing in the traffic streams there?

So that's really the foundation. And that's continued, I mean, since I got involved with this. And that's really been what we've tried to get-- we've tried to tune it more than rethink it, I think. And so how do you get better signal on smaller populations quickly? I think is really what we're after, so that we can have confidence that what we're shipping is correct faster. Right? That's really the story there. So I think what we say in the book is pretty accurate still, especially if you look at the macro level.

STEVE MCGHEE: So it's about precision, right? It's about getting finer and finer granularity, and being able to measure smaller populations, but also a smaller portion of your product. The one big thing I've learned at Google is that a small percentage of a large number is still a large number. And this is the problem when you have a large denominator, when you have billions of users in your product, is the smallest little thing is affecting like, oh, I don't know, two million people. Like, that's not great.

DANIEL HOBE: Right. Right.

STEVE MCGHEE: So how do you go about attacking that problem in a way that isn't maddening? Are there any tricks?

DANIEL HOBE: I don't know if I could say it's not maddening. But I think it is trying to-- you're throwing computers at it. How do you look at all of these different populations that you're able to track? Are you seeing that as you're rolling out this new feature, you're just getting much higher error rates in Japan or California, right? Are you clustering these things? And those are typically easier to find. And so that's kind of the direction we've been going.

But getting more fine grained, one of the related things that Search does a lot to roll out features and test features is these experimental changes that we make. So as you're a user of Search, you'll come to do a search and you'll get a slight variation of an experience. And then we see, did you like that more? Did you like it not as much?

But then we have sort of a control population, experiment population. And you can see just from an availability and error rate perspective as well, is there something wrong in the code that we're shipping for this experiment and those people are seeing more errors? Like, let's roll that back. And so it's something that we're trying to automate always more and push this out further. But it's a never-ending task. And it's a prioritization discussion. How far do you push it?

STEVE MCGHEE: So the way that I think about this, tell me if this is right, is you're moving away from measuring from the server side where it's like, this server is hot or not, and it's more like, you can measure the tranche of users, regardless of what server they're on, or even build, or even configuration.

You're saying, this set of users, which might be like-- this would be a bad idea, but you could have a tranche of one user. That would be the worst-case scenario, where you're like, experiment on this one person. But the point is that you're disassociating your measurement from the infrastructure that you would previously measure with. You're saying, spread it out.

DANIEL HOBE: It's an and rather than an or.

STEVE MCGHEE: OK. OK.

DANIEL HOBE: And yeah. If you have one person in an experiment, you're getting zero data. So that's just not-- I don't think the system even allows you to set something up that way because it's useless.

STEVE MCGHEE: But you wouldn't deploy one server to one person in the-- that would be ridiculous. But you could potentially have a very small user experiment group that doesn't map to any piece of hardware, I guess.

DANIEL HOBE: Yeah, I mean, so the most common one would be a team [? food. ?] Or an internal-- like, its launch to Googlers. And so a fairly small group of people, all things

considered. And we can see pretty early what the results are.

MATT SIEGLER: So this is cool. This sounds like shipping product, launch internally, test it, trusted testers, then super careful canarying, or quiet launch, shadow testing, all the stuff that we've been practicing, experimentation, A/B testing, rollbacks-- or not rollbacks. We haven't gotten that far. Then maybe launching it on a single machine, see what happens, and canarying it in a single cell, or in a locale, et cetera. I mean, just the world knows how this stuff works.

DANIEL HOBE: Just progressive rollout.

MATT SIEGLER: Yeah, yeah. Right. And then we watch metrics and we make sure that they behave properly on calendar time, respecting holidays, all the stuff that production does. But I hear this new stuff has been happening the last few years where we're launching things that aren't just software. They're also, like, big piles of numbers that do stuff that are pretty cool. So how is that ultra--

STEVE MCGHEE: I don't get it. What are we talking about here?

MATT SIEGLER: We're talking about AI. So we're talking about models and AI, which behave really differently from production. Their behaviors are super different. And we're launching them, I think, in different ways. Now how has that affected the way we observe their production? And we launch differently.

STEVE MCGHEE: Just use the old model, Matt. This can't be that much different.

MATT SIEGLER: I'm sure we're doing it differently. What's going on? What's different? What new kind of instrumentation, tooling, and rollout, production and monitoring and all the other stuff? We're doing it different.

DANIEL HOBE: Right. I mean, I think there's the technical challenge of that. But I think the other thing that's worth calling out is, I think there's a business difference as well. So Search is an established product that people kind of expect dial tone reliability from. It's always there, it always works. I depend on it. That's what we need to live up to. Especially with new things where the market isn't really well defined or it's still evolving, and anyone that's been involved with this will know, tools and approaches are changing week by week.

And so it's a lot of focus on going faster, trying new things. Are we finding that product market fit? All of these kind of things. So even just setting aside the technical differences, I think the biggest change is, as you think about rollouts and how fast you want to roll out, that's really a risk discussion. How much risk are you willing to take? And the math is different for these different businesses. So Search continues to be a little bit more conservative in making sure we've got all of the I's dotted, T's crossed, all of this kind of stuff before we go further.

Different parts of the company or different parts of their lifecycle are going faster. So some of that is same process, just push the throttle higher. And your canaries run shorter, you have less progressive rollouts, this kind of stuff.

So I think that's sort of step one is, like, we've just decided to take on more risk with this as far as reliability goes. If we have a problem we detect after it's fully rolled out, either just roll it back, and we're willing to take that on.

I mean, I think the second part of it is, we are reusing a lot of the tools. Can we canary things to trusted testers? We have a new model we're trying out. There's a whole process for this, both internally, team [? food, ?] we get feedback. And that thing, that life cycle, looks similar. But I think the end of it is, one of the challenges that we really see-- and this is industry wide-- is that these models are not really deterministic. So do you have a golden query set? Like, I put this in, I get this out? Well, not really.

So how do you know that is the thing working? So if it's not throwing errors, like, is it actually doing what the user asked? This kind of stuff is a huge challenge.

STEVE MCGHEE: Semantic errors versus syntactic errors are difficult to measure. Yeah.

DANIEL HOBE: Right. And so we've had this issue a bit in Search as well over the years, of it's a quality problem. Did we return the best quality or did we turn a crappy quality to a user? Everything looks green from a error count perspective, but the user didn't get what they wanted. And so that's a reliability problem as well.

STEVE MCGHEE: Totally.

MATT SIEGLER: This is great. How do you see SRE, with its mindset or background or its latitudinal approach, participating in the business decisions that allow making informed, risk-oriented decisions? It sounds exactly like what is happening, now that we have more uncertainty, we're doing new kinds of products that are different in completely substantial ways, but also higher rates. What is SRE bringing to the table that's uniquely valuable?

DANIEL HOBE: I think it's a few things that I've seen. So I definitely, sort of latching on to what you just said, I do see SRE as that advisor and partner, especially when you get to some of the more strategic discussions of, like, here's how we see the risk space of what we're taking on. As someone sort of an expert in the field, I think these are fine, this one makes me nervous. Let's have a plan for that. But at the end of the day, I want to sit down with the business owner and say, it's their decision what they want to do, providing them the information to make that informed choice.

So for example, we're doing a migration within Search from a parallel piece of infrastructure back onto more standard stuff. And everyone wants these launches yesterday. So we want to go quickly on this stuff. But really, the quality and the reliability weren't there yet on the new system. So as we walk through these launch reviews, I turned to the business lead of this and said, if we do this, we're going to take a reliability hit. We'll finish the migration, but users will see less reliability for their product. Is that what you want?

And I think the thing that sealed that discussion was, we took it away from talking about SLO percentages and saying like, well, it's 0.5% less reliable, and started talking about the number of impacted humans that would see this problem. So to what, Steve, you were saying earlier, talking about it in 500,000 people a day will see an error when we do this before. Yeah. Like, oh, I don't want to do that.

STEVE MCGHEE: [INAUDIBLE]

DANIEL HOBE: So turning that into, OK, here's what we actually need to button up before we finish. And it pushed out the launch a little bit. But that was the call that made sense.

STEVE MCGHEE: Sometimes it's a little hard to live in the high-abstraction mind space. And then someone's got to kind of pull you down to reality a little bit, like, this is what this means, like really, really.

DANIEL HOBE: Right. And I think that's something that we've done across, I mean, certainly both Search and Geo, Google Maps, has been to change how we talk about outages. So we do the whole postmortem process, which we talk about. But in that is an impact assessment. And I think a lot of teams were looking at a graph and being like, well, we had 3% errors.

STEVE MCGHEE: 2 pixels.

DANIEL HOBE: That's the impact. Yeah, 2 pixels of errors on the graph.

STEVE MCGHEE: Yeah.

DANIEL HOBE: And we've shifted this so that it has to be as close as you can get to an actual number of people impacted for this outage. And that's, I think, made the whole discussion a lot more visceral and real for people. Like you said, these large numbers are hard to wrap your head around when you talk about percentages. So I think that's been helpful, too.

MATT SIEGLER: How do you see what you've learned about AI-oriented launches, and how it's propagated backward towards more traditional software production practices? Because it doesn't just go in one direction. Presumably we've discovered some new things along the way

that inform generic principles.

DANIEL HOBE: It's a good question. I think one of the interesting things that I've seen sort of comparing-- and go back to these products that are in different parts of their life cycle, one of the things that-- everyone's always short of chips to launch these models. There's never enough chips. Everyone wants more. We want to get more traffic, all of this kind of stuff. So you really have to have a crystal clear prioritization of what you care about the most and what you're willing to drop.

And that's been a really great lens to look at this stuff with those business leads. We can do X or Y or Z, or some combination thereof, but you need to tell me, this is what I care about the most. This is what's going to drive whatever goals we've set for the business. And just really nailing that down has, I think, made the discussions, and even just the incident response, much more straightforward.

Because we have a very clear-- we're having some percentage of errors across everything. We care about this feature more than that feature. We're going to prioritize that. That's sort of a clarity that brings to the discussion that is sometimes difficult to get to. But because we're in a scarce environment of chips, or pick whatever dimension it is, we can have those discussions in a much more grounded way with the leads.

MATT SIEGLER: This takes me back to a recent episode we had with Matt Zelesko, head of SRE. And he effectively said, if I could paraphrase, which is that it's not all about the SLOs anymore. We got a bit in the weeds on that. That's what happens when you develop a theory of this. And after many, many years, we're like, we need to step back and think a bigger picture as SREs used to from the beginning, which is business orientation is actually delivering for the customer or the product, or wherever you want to think about it.

And I think this is what you're saying, is kind of stepping into the earlier part of the design, and delivering what is necessary on the business. I really like how you phrase that.

DANIEL HOBE: Yeah, I mean, I think you go way back, and one of the only somewhat joking things that the SREs would say to the devs would be, like, why do you hate your users?

Because you're launching things that aren't available and this kind of stuff. And I think that's always been something that I've gravitated towards and tried to position the SREs around understanding the user impact, understanding what the users are actually experiencing so that we can inform the decisions we are making with that lens.

And I think SRE is uniquely placed there because we do have that horizontal, bigger picture, more holistic view of things. And I mean, that's always been why I've stuck around the consumer side of the business, is that that to me has always been rewarding.

STEVE MCGHEE: Backing up a little bit with regards to where Search and AI meet together, we have this product called AI Mode. And it's a part of the Search world. I think any listener who's been aware of this whole space understands that AI models have to be trained, and then there's inference, and there's reaching out to data sets where things like RAG, and stuff like this. But I guess my question is like, this all happened pretty fast, and your team had to keep an eye on this new system in this old-- not old paradigm, but in this existing framework.

Did you have to run two ways of looking at the system in parallel, like classic Search and new Search or something like that, and have two teams, and two dashboards, and two rollout, or n? Maybe it's more than two. I don't know. It sounds like a lot of work, is what I'm saying. Like, how'd it go?

DANIEL HOBE: It is a lot of work. It was a lot of work. And I think down to a huge effort, especially from the Search SRE teams, to get this out the door.

STEVE MCGHEE: Yeah.

DANIEL HOBE: But especially early on, where even prior to AI Mode, we had AI overviews,

where we're experimenting with this stuff.

MATT SIEGLER: Correct.

DANIEL HOBE: And we know we need to iterate quickly because, like I said, we're finding product market fit, we're getting user interactions with this. They're figuring out how it works. We're figuring out what works for them. And we kind of need to continue to grow, and grow and change the product.

So really, this was, how do we-- again, fairly typical. But we can stand up a more isolated part of our infrastructure where those queries go to. That means that if we push quickly, and we have less progressive rollouts in that part of the stack, the blast radius is contained to just that. So I think the approach is, when you talk about it at a high level, relatively standard. I don't know, it feels intuitive to me. But I think the mechanics of actually making that work in a system as complex as Search is where the difficulty was and the effort was. And I think the SRE team really delivered that.

STEVE MCGHEE: All right. I've got an oddball question for you. So prepare yourself. You were not briefed on this.

We've been around long enough where we have friends who have moved on to other companies, and are out in the world, and we chat with people on, I don't know, LinkedIn and stuff. And for a while, my job was to go talk to, specifically, Cloud customers. But it was like, people who are doing this role, like the type of people who listen to this podcast are like SRE adjacent, like platform engineers, and devops, and platform teams, and software engineers who just happen to run prod or whatever. And TPMs in the same world, stuff like that.

So there's a lot of folks that are out here doing things at many levels of, whether it's scale, or newness of a company in many different industries, stuff like that.

So my oddball question to you is, if you were to be magically transported into one of these teams out in the world, like, just pretend for a minute that you know what you're doing in whatever this company is, what would you bring with you? How would you manage your second mover advantage, where you're like, I've seen all the stuff that Google did, but I'm not at Google now. I'm at, foo.corp. What would you take with you? What would you be like, OK, step one, get blah? And purely theoretical, not in--

DANIEL HOBE: Yeah. I mean, I think, before I answer the question, I think we've kind of seen answers from other people that have left Google and then turned around and were like, well, I missed that piece of technology or that infrastructure. And now there's a bunch of open source for it.

So I think there's a lot to be said for the infrastructure and what it's capable of. I think it's also very complex, and there's a lot of knobs that we have. And so I am thinking about this. I think sort of echoing what we just talked about, I think what I'd really want to bring with me is some of the high-- or low-latency monitoring and data analysis that we have for our systems.

So we get a ton of information and try and build observability systems for our products. So I think that's something that we rely on a ton to understand if it's working. And I think bringing that out to the edge, I think we actually do a fairly good job of understanding how our infrastructure, or our binaries, our systems are working. Tying that all together to, how is the user experiencing it, I think we've made a lot of improvements to. And I would love to make that a first-class citizen almost, or however you want to describe it, of wherever we go to next, because I think that really tells that full story of what you're trying to do.

STEVE MCGHEE: Yeah, I have this theory that a real job here is not just to make Google better, but it's to make all software everywhere better. Because Google can be great at a bunch of things, but my hospital is still my hospital. And I hope they have good software too, or the post office, or whatever. And I don't think Google is planning on running the post office anytime soon,

but it would be really nice if the team that is running system X is able to kind of maintain their systems and observe them.

So I like what you're saying there. So it's about quick feedback time. You refer to latency. So when it comes to a thing happens out in production, how quickly can you see that it happened, at what fidelity? And then when you talked about to the edge, I think you're talking about kind of the breadth of your deployment, and making sure that you have that same fidelity and that same speed across all, I don't know, locations, potentially.

DANIEL HOBE: Locations for sure. Or if you think about in the consumer space, actually being able to tie what's happening in the app on the phone, the user's interacting. They want something from search, and they're trying to get to it. Or they want something from maps. Is it working? Is it doing what they wanted? And can we tie that back into a change that we made that's now causing an issue? How do we fix that for them?

STEVE MCGHEE: Awesome. Cool. Well, I think we covered a lot of ground in not much time. I think that was actually pretty amazing. Before we leave, we always like to ask folks, where would you like people to see you on the internet? Do you have a place where you write down words that you want people to read or anything like that?

DANIEL HOBE: Not really. I'm--

STEVE MCGHEE: The backup joke is always, Strava. People can just see where you ride your bike. That's what I always recommend. But--

DANIEL HOBE: That is probably where I'm most active on the internet right now. So Strava or maybe you catch me out in the Marin Headlands on my bike. I've got a GitHub that I'm personal projects and messing around with stuff on. But yeah, don't have a huge presence out on the internet. But yeah, no, this has been great. Thank you.

STEVE MCGHEE: Cool. Any parting words, Daniel or Matt, before we hang up?

MATT SIEGLER: We like to hear pearls of wisdom maybe outside the specific area we've interviewed you.

STEVE MCGHEE: Yeah, no pressure. Just-- [CHUCKLES]

DANIEL HOBE: There's been a huge amount of discussion, I mean, internally and externally, around AI models, and what they mean for both the discipline, and the industry at large, and all of that. And I think there's a huge spectrum of what people are thinking about here. And I think there's a lot of concern, and worry, and anxiety about this. And I think a lot of it is, it's fair. It's a big change. And I think that's sort of a natural reaction to it.

One of the things that I think I am trying to tell the story of how we use this. And I think one of the challenges that SRE at Google, in particular, has had is, our environment is hugely complex. Just if I think about what Search used to be when I started and what it is today, like, I'm not sure there's more than a few people that could draw a full whiteboard of all of the different pieces.

And I think that complexity has limited what we've been able to do. Just trying to get all of the pieces moving together to do something that affects a larger area of production is harder. And I really see these as, this is an opportunity to use those tools in a way to enable you to get back to what we used to be able to do.

So I think that's hopefully maybe more of a positive outlook on this because I think really, what we continue to depend on all of our SREs for is that human judgment, what is worth doing. And I think a lot of things, you talk to anyone that's been around for a while, they could rattle off 10 things they know aren't great, they know need to get fixed, but either it's just too much of a hassle, it's too much to get it all done, too much to clean up all of this.

And I really do think we can lean into these tools as a sort of engineering enabler to tackle some of this stuff. Which doesn't mean there's not negatives and downsides that we'll work through and all of that. But I think that's a way to get back to more of what at least I want SRE to be.

STEVE MCGHEE: Well put. You mentioned the whiteboard. There was a whiteboard downstairs at Google that was very impressive, very big, full of many things. It was very funny. There was a lot of jokes in it. A lot of was real. Some of it was silly. It was written by a guy named Cody. Cody has been on the podcast in the past. You can find videos of Cody all over the internet, but we lost him recently. It's been very sad. So we miss you, Cody. I know you worked with Cody a lot too. So--

DANIEL HOBE: I did. Yeah, no. He was actually in my wedding party. So-- knew him well.

STEVE MCGHEE: Well, thanks very much, Hobe. That was a weird way to end the podcast. Sorry about that.

MATT SIEGLER: It's personal.

DANIEL HOBE: Yeah, thanks for having me. Catch you around.

MATT SIEGLER: Thank you.

DANIEL HOBE: Bye.

SPEAKER: You've been listening to *The Prodcast*, Google's podcast on site reliability engineering and production software. Visit us on the web at sre.google, where you can find books, papers, workshops, videos, and more about SRE. This season is brought to you by hosts Jordan Greenberg, Steve McGhee and Matt Siegler. *The Prodcast* is produced by Paul Guglielmino and Salim Virji with contributions from many SREs behind the scenes. *The Prodcast* theme is "Telebot" by Javi Beltran and Jordan Greenberg.

["TELEBOT"]