

[JAVI BELTRAN & JORDAN GREENBERG, "TELEBOT"]

JORDAN GREENBERG: Welcome to season 6 of the *Prodcast*, Google's podcast about site reliability, engineering, and production software. This season, we met with SREs in person to hear what's on their minds, to explore the importance of psychological safety, and to learn what's coming next for SRE. And of course, the most important part is the friends we made along the way. Happy listening, and may all your incidents be novel.

MATT SIEGLER: Welcome back to the podcast, Google's podcast on SRE and production software. I've got a special guest in person today. I'm Matt Siegler. This is--

MATT ZELESKO: Matt Zelesko.

MATT SIEGLER: What are you? What do you do here?

MATT ZELESKO: I lead SRE teams for Google globally.

MATT SIEGLER: How long has that been now?

MATT ZELESKO: It's been a little over four years.

MATT SIEGLER: That's right. And you were here just 10 months ago.

MATT ZELESKO: Yes.

MATT SIEGLER: And the SRE book is now 10 years old.

MATT ZELESKO: It's amazing.

MATT SIEGLER: And in 10 months, so much has happened. What has happened in 10 months? Let's start with you.

MATT ZELESKO: Yes. Well, first of all, the SRE book is 10 years old. We've had small things like the invention of cloud and hyperscalers and now AI that have all happened after the book. So I think we're really mindful that the industry and the landscape has changed a lot. And we need to change with it. What has happened over the last few months is a real shift in how we interact with AI and large language models.

This was generally a dialogue-based interaction. You were talking to the model. You were chatting with the model. It has now shifted over to agents and agentic workflows. And that's really only been over a few months, but we are already seeing really dramatic impacts as a result of that. And I'll say, and I probably said this 10 months ago, it's hard to guess how fast this is moving. But I think it's very clear that the role of software developers is changing and changing right now. And the role of SREs is changing and has to change right now as well.

MATT SIEGLER: I like how you put that. Last time you were here, you called AI the buddy, the buddy sitting next to the SRE, maybe like in the passenger seat, but now it's like the buddy's become a lot more autonomous. Let's reflect on what SRE's role is in this increasing velocity of that buddy now. What's buddy doing now?

MATT ZELESKO: Yeah, so I think you're absolutely right. It's going from human-centric work to human-supervised work in a lot of ways, which means that buddy or actually, it's like buddies that are doing a bunch of work on your behalf. And what this means is that the pace at which we are creating code, the pace at which we are changing production, is increasing dramatically. And that changes a whole lot of the stuff around the role and what we have to do as SREs. And look, I always think that there's going to be a human element here. We have a judgment and oversight and a wisdom that is still very much required. But a lot of the other work can be done by agents. And we're seeing more and more of that happening.

MATT SIEGLER: Yeah, you mentioned last time we talked SRE will always be in the operations role, and AI is just shifting what our operations role can do, maybe changing the scale, maybe changing when they take action.

MATT ZELESKO: Yeah.

MATT SIEGLER: That implies a lot of complexity.

MATT ZELESKO: It does.

MATT SIEGLER: Increased complexity. I know that SRE role and mindset is to be comfortable in that complexity. Do you think that's a role suitable to this new world that we're entering into? I think that's what you're saying, but--

MATT ZELESKO: I'm saying a few things there. I think, number 1, you are absolutely right that there is a notion that SREs pride themselves on having really deep domain expertise. And I think in the future, we are going to prioritize generalist capabilities. So I think we'll still have expertise in areas, but I think we're going to really emphasize this need to be able to generalize across a bunch of different domains, to not be as restricted in terms of the expertise we have today.

At Google, we have really deep domain expertise simply because the scale is massive, and we almost have to. But I think as agents take on more and more of both the software creation and the production operation, we start to expand out and we each need to have a lot more broad domain expertise as opposed to specific.

MATT SIEGLER: Yeah, it's this expertise point. It's interesting to me. Let's do some before and after, like zip back a few years. Not all the way back to the beginning, but--

MATT ZELESKO: Sure.

MATT SIEGLER: --enough back in time where it felt like expertise could be focused and you could still get your job done and that would be enough. But I think you're talking about maybe broadening one's experience. Let's think outside of Google, maybe industry as a whole. What would you like people to be skilling up, maybe? What they need to be doing to prepare for this?

MATT ZELESKO: Yeah. Well, one thing I would say is we've already seen a transition like this inside of Google, and we may have talked about this on the *Prodcast*, but there's been a real concerted effort over the last five years to get on to common production platforms for things like rollouts and observability, capacity management. Incident response. Because prior to that, we had many systems that did those things, and there was real power that was unlocked when we said, hey, everybody's going to get onto the same systems.

That, in and of itself, helped us expand the knowledge base of the typical SRE and the systems that they could actually manage. So if everything is running on the same production platforms, that means if I'm an SRE in one area supporting YouTube, I can go and support Workspace because I understand all the platforms, I understand all the tools. So we've already had a model here where we've seen a real uplift in our ability to be productive and productive across multiple domains when we start sharing common tools. And I think this is another example of that as well.

In terms of what I would suggest somebody in the industry does, start using these things as quickly as possible. It's what we implore our teams to do, and so what we've really encouraged is start using this thing. Just use it in the things that you do every day. And there's a responsibility on us here, leadership in saying, we are going to grant you the time and space to do that. We understand that adopting anything has a bit of a productivity dip at the beginning. And then you presumably get a lot more productive. And so we've got to give the team space to go and explore and experiment with these. And I think we also have to provide the team with safe and useful default ways to do it, but particularly when you're talking about managing production.

MATT SIEGLER: Yeah, I hear that, especially this exploratory, it does feel like although we want something right now. We're still trying to figure out what the thing is, and the thing keeps changing. How are you helping your teams figure out how to control the variables on this? Like, explore this, but also you still need to get your work done. Try this out, but also you still have to

roll out safely.

What are the parameters here? How do other engineering managers and their directors and their CEOs make good decisions about letting that into their system, but also confine it so they can explore this? Because they know they're going to have to deal with it later. And they'd like to adopt it, but not all want to go. What's the strategy for this?

MATT ZELESKO: So I think there's a bunch of different aspects, and we've actually tried to break it down into some of the different parts of the role and how you apply it there. So when you think about making changes in production, you have the rollout and supervision around that rollout. And then something goes wrong, so you have an investigation phase to try and understand what's going wrong. And then you have a mitigate phase, where you take actions to actually fix it.

And so we are thinking about each one of those elements separately. And so investigation is something that you can do relatively safely. You're not mutating production state. You're not doing anything else. And so we want really broad experimentation and adoption with investigation, having AI-assisted investigation. Mitigation, where you're actually going and changing stuff in production, we definitely want a human in the loop at this point. And we also are fairly limited in the types of things that we're going to allow agents to do in production today. So we're trying to take different elements of the job and the role and say, here's where we think you can really lean in, and here's where we've got a lot more work collectively to do in order to make those things safe and reliable for managing production. And I really mean "we" in the sense that I don't think it's a central team that builds these things. I think it is really-- everyone in SRE figuring out, what are the right sets of skills that work for them, and then starting to learn and generate common skills that everyone can use as a result of the learning that we have on the ground.

MATT SIEGLER: Yeah, I like how you called attention to this common skills, because one of the things that SRE as a concept, even internally, externally is this mindset of being able to really be down in the body of the work and having intuition for, oh, I think this reminds me of something that bothered me yesterday. And that kind of skill, that deep intuition of this deep in the stack and this concern that letting more and more automation alone, automation, not just AI, take hands on the wheel will blur if not deaden our ability to this intuition.

What do we do to preserve this intuition that we still want the hands on the wheel on occasion, because we get too much automation? Eventually, we'll be unable to address the complexity. We'll be, oh, it's reached the state, I can no longer address. The machine is handed back to a human in this loop and they go, actually, I don't have any experience here. I don't know what to do. We don't want that to happen. So how do we keep these skills sharp?

MATT ZELESKO: Yes, so you definitely still want to retain knowledge about the production infrastructure. As I said, that may be abstracted up a few levels from where we have that knowledge now. And that's, I think, OK. That enables us to look at a much broader landscape. So you still want to preserve some of that. I actually think that AI can help us with that because it can teach us about what the architecture is. It can sort of keep a repository of the information that we need and have it accessible whenever you need that. So I think there's actually ways in which AI can help us in being better experts at those things.

MATT SIEGLER: Yeah, so actually, there's some tension here. Role clarity and role specificity is something that SRE has claimed to have for quite some time now and has actually identified itself. And going to conferences, I was at SREcon a few weeks ago, it was very delightful to see people celebrating that, and also the idea that you can put someone on incident response, even in another company and with a little bit of time actually makes sense of what's going on. And that, as you mentioned earlier, even within teams or across projects, is an incredible

superpower for an organization.

At Google, that is certainly true. But as you describe this sort of proliferation of different kinds of production surfaces, maybe everyone has their own unique way of solving the problem, different kind of software stacks. That's a return to the way things were before, and I don't think that's necessarily what we want. Maybe in the service of innovation, that would be delightful. But how do you see working this tension out? Like, what are we going to do?

MATT ZELESKO: Yeah, I mean, we're talking a lot about the risks, but I also think about, what are the huge opportunities here, right? And I think here's some of them. There's a lot of times that we choose not to refactor code or rewrite code, because we just understand what a giant task that is. And it could be something that would help the reliability dramatically, but we're just not willing to invest in it today. I think that is going to change. Like, our ability to rewrite code, refactor code is going to increase dramatically.

I think the same thing, when we run something across Google, we tend to call it horizontals. We have an initiative where we go, OK, we need every team to adopt this certain thing. And we try and have a rule that if you have a horizontal, the central team, whoever's running the horizontal, is doing most of the work so that we're not putting work on all of the teams across Google that need to implement this horizontal. But the reality is all the teams wind up doing a lot of work. I think we have an opportunity here to say, if we ship a horizontal, we're going to ship skills with it so that every team can just use agents to do the work to satisfy that horizontal.

MATT SIEGLER: Remind us for those audience listeners what those things that may not be familiar with the skill of an agent, how they work together.

MATT ZELESKO: Oh, so skills are essentially just capabilities on top of the coding harness, whether we use Antigravity, which is Google's product. And skills are essentially very specialized capabilities that you can write. It's essentially ways that you can create new techniques or uses from just the base AI harness. And so give you an example, we have recently-- a lot of people have probers because it's a big part of SLIs and SLOs, and we want to deprecate one kind of prober and in favor of another kind of prober.

Historically, we would have gone to every team and said, OK, you've got to change your code, and here's the guide of how you should change it, and go and change your code to do that. Now, we can ship a skill that you just run on the code base, and it changes everything for you, right? So I think there's an opportunity to get things done and get things done at scale actually goes up really dramatically.

The other big thing I'm excited about is I think we aren't always in the room at the right time, and particularly in the room early in the process of designing a system. And so a lot of the reliability, knowledge, and expertise isn't always considered at the points where the architecture is being set or the software design is being set. Now with AI, I think we have the ability to essentially have things that are automatically comparing designs against our production principles and catching reliability considerations much earlier and much more upstream of where we tend to catch them today.

MATT SIEGLER: I was going to ask you that, but I'm so glad you brought me to that, because last time, you mentioned that SLOs were just a trailing indicator of reliability rather than leading. And how do we retool SRE for this role, which is effectively almost a product management role, which is defining by default reliability-- by default, I guess, is the expression?

MATT ZELESKO: Yes.

MATT SIEGLER: How do we get them earlier into the design phase and using these techniques that you just described right from the start? And you're saying your idea is to use AI to help actually indicate where they're missing?

MATT ZELESKO: Absolutely, because I think as we look at the agentic impact on software

development, it means that our software engineers are going to spend a lot more time defining the best spec they can for what they are trying to create, and the agents are going to create a lot of the code. And so when we are investing that much in that spec development time, there's a real opportunity to inject reliability and reliability thinking into that spec as well, in a way that I don't think we do today, or at least we don't do it uniformly.

And so, at the time that we are creating the specifications that our agents are going to use to go write the code, having additional skills that go in and look at the spec and assess it for-- identify risks with it, right? If availability is the trailing indicator, risks are the leading indicator, right? We try and identify the risks associated in a system. And we have some great mechanisms like STPA and other things that are ways for us to do that today. Today, that is fairly human intensive, but you can imagine risk identification agents that are just running all the time and looking at this and trying to find things in the code. So whether it's at spec time or even at commit time, they are going and assessing the production risk of the changes that are being made.

MATT SIEGLER: Yes, this is a very design-heavy conversation we're having. If I'm building an architecture or even revising a very large foundational architecture or moving along its life span, I imagine there are large architectures that have designs that are heavily departed from where they are in reality right now. And that's just what happens. That's life, right.

MATT ZELESKO: Yes.

MATT SIEGLER: And this is an opportunity to go back and see what's happened and maybe go identify that as an opportunity for revise. Now is that happening now or are we actually going and finding out, oh, this is not what's really going on anymore? In fact, we've just discovered this? This is an AI opportunity?

MATT ZELESKO: I think it absolutely is an AI opportunity. And it is one where to your point, it's not a point in time spec, but it's ideally like a living diagram and understanding of the dependencies. And I talked about this before a little bit. But I think dependency mapping in this new world, particularly with much larger numbers of services, much larger production surface area, you've got to have a real time understanding of the systems and how they work together and how they depend on each other, and I think AI can do a lot of that for us.

MATT SIEGLER: Yeah, I'm going to be repetitive on this point, but it is, of course, the *Prodcast*. Why again, do you think SRE as a collective, as a culture is especially suitable to this role? We have so many development engineers, and SRE has been training for this particular job, apparently. And why are you trying to post position us for this? It seems like you are. Tell me why you think that we are culturally suitable for it at this moment. Like, make you--

MATT ZELESKO: So a few things I will say, number 1, and this went back to the original SRE book, SREs' job is to automate ourselves out of a job. It has always been about identifying, What are the difficult manual steps, the toilsome steps? and essentially automating them away into the next thing. And so when we say automating ourselves out of a job, it takes on a different tone today, I think.

MATT SIEGLER: I think.

MATT ZELESKO: And so we've been talking a lot more about automating ourselves into jobs. And I think in that case, it is all about, what are SREs the best at? They're best at reliability engineering, having really good understandings of how to build reliable, resilient systems. And, yes, we also do a bunch of toilsome work. And we are on-call, and those are aspects of the job. But I think at its core, we are reliability experts. And in any agentic future, you still need reliability experts.

And so in some ways, I feel like this shift to agents and agents writing software is just the next step in the automation of a lot of the work that we do. And SRE has been doing that from the

beginning. We understand how to identify toil or tasks that are ripe for automation and then going off and doing that. And I think that's going to be the same and very true over the next 10 years as well as we look forward.

MATT SIEGLER: I think you pointed out some areas of production that even a small shop can take action on, especially in the anomaly detection and mitigation, not necessarily in the rollouts, but these are things that even the smallest shop can do right now.

MATT ZELESKO: Yeah.

MATT SIEGLER: What are the things that are surprising you right now that you've seen happening around the world? What is something that you did not expect that is happening?

MATT ZELESKO: I did not expect how fast we would make this shift. And I suspect I'm not alone there, but just the pace at which we went from, as we talked about 10 months ago, AI as the companion helping us out to AI really taking the lead role in a bunch of the work that we are doing today. And I've been frankly shocked by its capabilities. I was a software engineer. I think we actually even talked about that on the first *Prodcast*, but I hadn't written code for a while. And as I've implored the team to spend more time with these things, I spent time with the system. I went and picked a bug that had been sitting there for a while, and I worked with Antigravity to go through and fix that bug. And it was probably the first time that I had created software in a really long time. And what surprised me was how good that collaboration and interaction was and how good the results were that came out of it.

And also I got a little bit of that sense of wonder and fascination with things that are possible. It really opens up like a whole new world of possibilities. And I think until you start to play with it, you don't quite understand how amazing it is.

MATT SIEGLER: And how about where are you going from here? Like, some places that you dream big? What's something-- extrapolate your way out. What would you like to see in 2028--

MATT ZELESKO: Well, look, as I said before, today, we are looking at, what are the safe non-mutating parts of production where agents can have an impact? I think that is a period of time. And if we do project out. And so you step back from that and say, generally, SREs are going to be on-call a lot less than they are today.

And why that's exciting is because we have structured so many things around SREs being on-call, even down to the way we structure our teams. So if we create a team, an SRE team for a service, that is two sister teams and two geographically distributed locations, so we can do around the clock very easily. What does this unlock for the organization if on-call isn't the grounding for so many of those decisions we make?

That starts to get really exciting, right? And it doesn't mean that we aren't still the stewards of production, and we aren't responsible for the reliability and resilience of the systems, but the way that we do it, I think, can be a lot more empowering.

MATT SIEGLER: I like the sound of that. Maybe we won't get so many pages at 2:00 in the morning.

MATT ZELESKO: [LAUGHS]

MATT SIEGLER: All right, well, that's all the time we have from the Pittsburgh office of Google. Matt Zelesko, I'm Matt Siegler for the *Prodcast*, Google's podcast on the SRE and production software. Thank you for joining us.

MATT ZELESKO: Oh, happy to be here.

JORDAN GREENBERG: 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, Florian Rathgeber, and Matt Siegler, with contributions

from many SREs behind the scenes. The *Prodcast* is produced by Paul Guglielmino and Salim Virji. The *Prodcast* theme is *Telebot* by Javi Beltran and Jordan Greenberg.
SPEAKER: You missed a page from Telebot.