[JAVI BELTRAN, "TELEBOT"]

SPEAKER: Welcome to season five of *The Prodcast,* Google's podcast about site reliability engineering and production software. This season we are continuing our theme of friends and trends. It's all about what's coming up in the SRE space, from new technology to modernizing processes. And of course, the most important part is the friends we made along the way. So happy listening, and may all your incidents be novel.
[STATICKY VOICE]
STEVE MCGHEE: Hey, everyone. Welcome back to *The Prodcast,* Google's podcast on SRE and production software. I'm Steve McGhee again. And also again, we have Matt. How's it going, Matt? How are you?
MATT SIEGLER: Hey there, Steve. Glad to be back.
STEVE MCGHEE: Excellent. I think we have a guest today too, don't we? Probably.
MATT SIEGLER: We do.
STEVE MCGHEE: We tend to. Yes.
[LAUGHTER]
Guest, would you please introduce yourself. Who are you exactly?
DENIA DEL CID: Sure, sure. My name is DENIA DEL CID, and I'm an SRE at Google. I've been in the company for about 9 and 1/2 years now, and an SRE for the large majority of it, like 7-plus years, 7-8 years.
STEVE MCGHEE: How did you get hired? Were you hired not to SRE, then you moved into SRE?
DENIA: Yeah, no, I actually started as an intern. I started as an intern working in support. If you are a Googler or were a Googler, I used to work at TechStop.
STEVE MCGHEE: Nice, classic.
DENIA: And then I did a rotational program that allowed me to travel around the world and then rotate with teams. And through that, one of those rotations, that's how I discovered what SRE was, and I loved it, and then here I am a bunch of years later.
STEVE MCGHEE: A bunch of years later. We don't need to count years. No one feels good about that. That's very cool. I've actually heard a lot of folks come through either TechStop or corporate IT. That tends to be a pretty-- I don't know if it's common, but I've heard it a fair amount of--
MATT SIEGLER: Yeah, the route through operations into SRE makes total sense to me, like getting your hands on things, then using that on the things to support the systems. That makes perfect sense to me.
DENIA: Yeah, back in my day, it was not as common, but I think you see a lot more of that, which is great. I think there's a lot of overlap for sure.
STEVE MCGHEE: Are you implying that I'm some sort of old person that's been around for too long? Because that's exactly accurate.
DENIA: No. It's the opposite.
[LAUGHS]
STEVE MCGHEE: OK, so I'm guessing that you're no longer in TechStop. What is it that you're doing these days? So what SRE flavor do you practice?
DENIA: So nowadays, I am essentially doing AI for SRE.
STEVE MCGHEE: Cool.
DENIA: Which is a very cool, very exciting, definitely fresh field. I've only been doing this for the last year and a half, shortly after the inception of my new team.
STEVE MCGHEE: But to be honest, a year and a half of AI for anything is like the half life of all

of AI at all anyways, so that's pretty good. It's a fast-moving world.

DENIA: Yeah, it changes so much. It does. When we started, Gemini 1 and 1.5 were still a thing, and now we're all the way to 3. So I've definitely seen the evolutions and the models changing along the way.

STEVE MCGHEE: For sure. That's cool.

MATT SIEGLER: That makes me think about SRE being cautious about adopting newer infrastructure into its folds because it's so focused on repeatability, reliability, and wanting to ensure the best for its services. Can you speak a little bit about that in AI and how, taking that into it, being cautious and thoughtful about it, that says something interesting about that, for SRE to say, OK, yep, we're going to do some AI. How did we do that?

DENIA: Yeah, so back in the good old days of two years ago, there was definitely a lot more hesitation. It wasn't too clear what the path of where the applications of using AI in SRE workflows would look like. And that's where my team was a pioneer in this field, where our org got funding for this exploratory project that is Data Cloud Platform, the team that I'm on, to explore ways in which we could apply AI, and then how reliable and applicable in terms of accuracy and to the field, whether it's to diagnosis or planning, how applicable AI could be.

And then now we're a year and a half later, since the design and the experiments and running the services, and we've been seeing a lot of progress both in terms of accuracy, a lot of interest in adoption, but it's definitely been a journey. I'd say what we've tried to focus on more is on making tools that surface information first.

And then, as you learn to trust your tool or to tune your prompts to the LLMs to get results that are specific to your team and that are high in accuracy, high being how a human in their own subjective experience would apply certain tags or would perform certain analysis, try to match that as closely as we can. That's where we're taking the journey. And then eventually, just recently, we've been starting to explore so now, how can we automate some of these workflows, maybe not fully 100% touching production, but definitely being that sort of companion to an SRE.

STEVE MCGHEE: Cool. Can you remember a moment where either you or someone that you heard about, or maybe this is just an experience or a story, where it went from, this seems maybe interesting to, oh, we should do this. This turned a corner. Do you remember any moment, like, what was the thing that it's like, oh, it can x, whereas before we were like, I wonder if it can x.

DENIA: We've had a couple of those. So for instance, one of the most interesting ones where we have all of these support cases and escalations that customers have mentioned. And then sometimes it takes a couple days for an OMG or an incident like an SRE to take a look at it, maybe because our alerting system was not aware of that niche case that the user was experiencing. Maybe we weren't monitoring it at the time. So that started as a very bare bones proof of concept. And then it started to be very interesting. We've detected a couple of outages, some that regular traditional automation detected, but some of them that were detected before in this method.

Another example is for leads across SRE. A common task that they have is analyzing all of their team's ticket queues and trying to find trends along those, like what was the most common root causes for them or the solutions? What proportion of the entire queue belong to a particular incident? Is there anything that we can automate and prioritize, because we're seeing high volume or because our team is dedicating too many human resources in troubleshooting these results? So we've been able to create tooling that analyzes all of that in a matter of a couple minutes, a task that could easily take some teams weeks, and then it just creates dynamic dashboards for them.

STEVE MCGHEE: Do you think it was a matter of the technologies themselves that you're building these on maturing, or was it that the team had to figure out which problems were solvable, or it was a matter of writing the agents or whatever within your team? Was it a practice thing or was it a OK, we're ready, now it'll work-- launch, launch, launch.

DENIA: So it was more of a hearing your customer kind of scenario. Our team is a horizontal across all of Data Cloud, which includes databases, data analytics across Google, which is a few teams, just because of the scale of Google. It's like tens of teams. So we sat down with all of them. A few people from our team came from those teams, so they already had knowledge about the common problems that they would have.

And based out of both their experiences and talking to the leads across these teams is when we started seeing patterns in trends or in workflows. Like, oh, your director every month is going to ask a report on what was the progress on x type of toil. And it was a manual task to have engineers.

Some teams, they would even dedicate a full-time engineer and give them a KR for the entire quarter to manually tag all of their bugs and then generate this analysis. So that's where we saw the opportunity of freeing up the time of an SRE as much as we could so that they could actually do the planning work, the design coding work that's one of the biggest values of an SRE, instead of having to read through all bugs and ensure they were tagged properly or manually tag them themselves.

STEVE MCGHEE: This is really striking a chord with me, just thinking automation-- like automation, automation, automation. This is our business model since day one. SREs try to automate themselves out of their own work. But usually with automation, when you go to your customer with an automation objective, they look at you a little funny, like, whoa, wait, whoa, I kind of want you to be looking at this manually before you hand it to them.

MATT SIEGLER: Yeah, it's important.

STEVE MCGHEE: --automate or-- yeah, my stuff is very precious. What are you doing?

MATT SIEGLER: Mine is better. It's more--

STEVE MCGHEE: Yeah, it deserves your personal attention. So let's think a bit about not only automating. You're now adding intelligent automation to it. So there must be occasionally some resistance or a little bit of skepticism.

How do you manage some of that? And how do you push through that, or maybe even back off from some of those things? And were there some pitfalls? Was there a negotiation through that process?

DENIA: Yeah, well, we do not force our users to adopt us at all. So if anything, what you're mentioning is exactly what we noticed at the beginning stages, like during our consultation phases.

So what we did is take notes. So we built the tool to be dynamic. So when people say, these are the tags that are specific to me, and this is how my humans tag them, and we're like, this is great. So give me your list, and we're going to make sure that when we analyze all of the bugs in your components, all of the tickets from your customer support cases, we're going to use your tags. So this is where AI, like your LLMs, are phenomenal, because we embed all of those custom tags that they have as part of the instruction.

The tricky part then comes with validation, because sometimes what is obvious to a tenured engineer in a team and common terminology for them might not be as relevant to the LLM. So especially in that very first cases after adoption, we notice people are dissatisfied with the results because they might not be accurate enough. So usually that takes a couple of rounds of prompt tuning, making sure that we clarify terminology or clarify different scenarios when a tag should be applied and when it shouldn't so the LLM learns.

And then we do validation. So for that, I do not trust this tool. Usually, teams tend to have what we call our golden data sets, which is what, back in the day, when they were still tagging things manually, we compare the results that the LLM handles against their manual results for the exact same bugs, and that's how we measure how accurate our labeling is against their tags.

STEVE MCGHEE: Yeah, that's really important. That's awesome that you had that, because without that it would be much harder, I have a feeling.

On the same vein, backing up a second, you're on a team that is, like you said, it's horizontal. It's kind of developing these tools or systems that are solving problems for neighboring teams. Is that about right? Based on your interviews your making the solutions match what they're used to, and so it's a little bit more-- I don't know, feels a little white glove, which is nice. It's not just like, go use this thing you've never heard of. It sounds like it's pleasant, which is great.

So once you have these solutions in place, and you're getting client teams to use them, how do you judge if it's working? What are the metrics for success that you, as this tools developing team, how do you know it's working? How do you know you're successful? Do you get points or something, or is there-- I don't know, I'm just curious.

DENIA: So it depends on the tool. So we have three main products for now. One of them is the early outage detection that I was mentioning through looking at support cases. The other is for analysis. And then the latter is similarity. So the way that you gauge success really depends. So for early outage detection, if you detected the outage before the team did, that is success to us.

STEVE MCGHEE: You being like-- if the tool detects it before the human--

DENIA: Before the automation.

STEVE MCGHEE: --gets paged or something?

DENIA: Yes, before a traditional SRE-alerting configuration would detect that as an OMG or as an incident.

STEVE MCGHEE: It's a race.

DENIA: Then that's how we qualify success in that regard. Then for the second is adoption of our dashboard in the regular workflows. So when teams tell us, oh, we loved your tool, we're now using it for planning, or we use it every week to analyze the trends of toil in our production reviews, that would be the second. And then for similarity is satisfaction with the results. So if we surface how similar a given incident might be to past incidents, so when those are high in accuracy, when teams give us feedback or when an individual gives feedback that it was successful.

STEVE MCGHEE: That's cool. Are you able to then roll those learnings back into the system to make it more accurate, more faster? I presume that's a goal, but does it work? Can you actually do that?

DENIA: Well, we're actually working on that right now. We're working on our pipelines.

STEVE MCGHEE: OK, that's very cool.

MATT SIEGLER: Yeah, I was curious how this talk about toil reduction. You are now getting some early signals, giving people things to look at specifically earlier than just hunting around their usual dashboards, which I'm sure they've cultivated carefully on their own. But now they might have some early signals that are actually maybe a little bit more specific to what's actually going on. Are you measuring toil reduction? And how is that going?

DENIA: Well, each team measures toil reduction independently, and they have different targets. Our team ourselves, we have relatively little toil, just because our services are so new. But from their end, they usually come with toil reduction targets. They have specific cues that have high volume of tickets. So most of the time, they want to reduce the volume of the incoming tickets. Other times, they just want to get better understanding of their tickets just because they are poorly documented.

And we do some context retrieval as well, where if a bug is related to a different ticket or to an OMG or to a postmortem, then we retrieve those contents to improve the tagging, which is similar to when a human would go in, read through a bug, try to figure out what happened, and label it.

So I think for us, we mostly set the general direction for the team. In the dashboards that we create, it's supposed to give them like a bird's eye view of what their toil looked like on any given period of time. It could be over the last year, over the last quarter, last month, last week. And we leave it up to them on what they want to prioritize.

We just surface the clusters that tend to be the most popular, just purely based on their own tickets. What were the most common root causes for the issues? And what were the most common fixes? The other interesting area is the ones that were-- the root causes are not known, that they might have auto-resolved. Why are you getting all of these tickets that ended up just being automatically closed or labeled as obsolete?

STEVE MCGHEE: Are you finding those to be fairly uniform objectives across your customers? They're all kind of targeting towards the same burndowns, they want to close tickets faster. No, you're shaking your head.

DENIA: No. Everyone seems to have-- they do have a goal of, we want to reduce our toil, but the how you reduce your toil or where in their journey they are to better understand their toil, it's all over the place. Some teams might have a better picture of their toil just because they dedicate so many resources to tagging and monitoring that through time.

And other teams are just starting to do that because now they're like, oh, we have AI that can do this automatically for us. That is so cool. Before, we were just overloaded with all of these alerts, we can barely stay over water, and they just feel like they're drowning. So this is starting to provide them with better insights, and it's taking them through the journey of should we remove alerts? Should we split some of them, or tier the alerts in how they get triggered? And others are going through the journey of, we should be documenting our tickets better.

STEVE MCGHEE: Yeah, teams go through these life cycles. And I would imagine that not everyone is going to use the system that you guys are building in the same way at the same time. It depends on context, it turns out. Context is important. Context-- who knew?

DENIA: Context is very important, yes.

STEVE MCGHEE: Speaking of context, how do you think that this process that you're talking about-- if you could magically transport it to be outside of Google, to be out in the world that doesn't have a borg, but instead has several clouds and Terraform and Kubernetes and words like that, does it still work? At the highest level, at the abstract level, say you transform it all to be working with whatever else is out there. Is this a transferable concept?

DENIA: The concept itself, yes. But our current tooling, I don't think it's a mature enough to be widely deployed. However, the concept itself, as long as the team that wants to deploy it has a relatively centralized or just a handful of data sources to pull from. So we're not looking, for example, specific board jobs and how they behave now. We're looking at what was reported in your tracking ticket after the alert fired. So if those systems are-- it's just one that it's ideal because then your integration is very easy. At Google, we do not have that. We report OMGs in one place, tickets in another, customer cases in another system. So it's just a matter of pulling--

STEVE MCGHEE: I think that's pretty common. I think a lot of teams are like that. But are you saying that the data that you're ingesting to do this analysis is like-- it's almost like second party data. Maybe it was originally meant for humans to look at, and you're ingesting it with an LLM. So it's the type of data where you're not querying an API and getting all the metadata from all the things, but it's more like, read the ticket that we have chosen not to read because they're super boring or there's too many of them or something like that.

DENIA: Yes, that is correct. There are efforts within Google-- I believe that you had Ramon, and they talked about the production agent. And then agents like that are designed to be more-- to analyze one ticket at a time. You're looking at an OMG and you're trying to analyze it and see what the current metrics are, what the current jobs are running. I think that type of integration with your job management services, that's more tailored to that.

What we are doing a lot more is like that post hoc analysis. So you resolved your bug. What went wrong? How did you fix it? And then for teams that are interested in also analyzing their incoming still open bugs, we also do that. We just do not surface details, like what resolved it, because it's still open. But at least we can highlight, oh, this was similar to these past bugs that you had.

STEVE MCGHEE: Got you. Cool.

MATT SIEGLER: Yeah, what's something fascinating about your input material is just language. It's just talking. It's humans meant for humans, although sometimes stultifyingly boring content for humans, so much that they don't actually read them. But that's the deal. It's like, here's a bunch of business data, but the LLMs have the patience of a saint, so it's just going to read them all and then learn things.

And I imagine there's some pretty surprising insights that come from reading all your things. And you're like, wow, you really figured that out. And wow, that's really interesting. So what are some surprising things that you've noticed that it can discover from reading all the human things that have been written about all the things? And it comes up with magic, surprisingly so. I imagine you have some anecdotes you could share, at least one or two.

DENIA: Unfortunately, I don't have as much as you would think, because their tagging is so discrete. Like I mentioned how teams can bring in their own labels, and then we apply labels from even on the default cases, that people are like, oh, just take my bugs. I just want to use-- I'll take whatever tags you have. They are still a very discrete set of tags. And then for privacy reasons, we do not store that contents of the original data sources in our tables. So we only get to see details of, it was this bug, and these were the tags that were applied because of so and so thing.

So we have to go back to the teams OK, these were the results. Do you have a golden data set? Let me compare. And in the cases where they don't, then you just learn about teams' configurations. I think that's probably, from a personal level, what I've learned the most is, oh, so pub/sub tends to have these two different queues. That is interesting.

And some of them, they tend to have this type of issues, whereas this other one doesn't tend to have that type of problem. Or in the context of, say, cloud pub/sub, then these labels are applicable, but these ones aren't. So it's more nuance like that, rather than getting the deets or the juice from what's happening in the system [INAUDIBLE] much of that.

STEVE MCGHEE: Well, so it sounds like your team is kind of attacking a pretty significant slice of the problem space that is SRE work. So I'm curious, let's presume that continues or you finished-- I know you can't finish, but presume that gets significantly improved. What would be next? So a bunch of people have thought about this, and there's a bunch of things you could possibly work on as an SRE or have help from an AI or an LLM as necessary.

So there's just writing code, there's dealing with incidents directly, there's root cause analysis of-- I guess that's part of dealing with incidents is actually fundamentally looking at what's going on-- and then there's design of new systems. There's all sorts of things that you could go to. Where would you go next if you wanted to do that? What do you think is the most either interesting or maybe the most fruitful and impactful? Where would you say it would be a good growth area to continue this type of work?

DENIA: It is interesting because I think, right now, the horizontal view, if I was in a regular team, and I would be thinking of, oh, I wonder how I can automate-- or reduce the toil-- some tasks in my team. I would then want to evangelize it and make it a horizontal effort. So our current goal is to help reduce the toil across this area at Google.

So I think the next stage where you can take that-- because part of what we do is not just surfacing the information, but we're starting to get into a space of deploying agents that rely on our information, because then you currently have teams that may ask, oh, this dashboard is amazing, but I want to understand why during this week we had so many incidents of this type. Can you tell me a summary about that? So we usually, when we get questions like that, we have to go back, modify the dashboards, or modify the prompt in some ways, which is maybe like an hour task, tops.

But by bringing agents, they don't even have to come to us. They can go straight to the agent and ask that information of the agent. We can customize it with custom SQL queries. But the agent also can come up with ad hoc ones displayed to the user, run them on their behalf, and then help them come up with that analysis. So I think the next part after doing this would probably see or try to see if we could deploy something similar to that for our external customers.

STEVE MCGHEE: This sounds to me a lot like a business intelligence role, but it's for production, which is, I think, pretty cool. It reminds me of a joke I saw recently, where someone was wearing a shirt or a hat or something and said, "as a kid, I always wanted to transform structured data into actionable business intelligence," which I think is pretty funny. Like, no one dreams about that. But it turns out it's actually pretty important. But now you're doing it in a more automated way, specifically about production, which is pretty cool, I think.

Cool. Well, thank you very much. This has been great. Is there anything that you want to make sure you get out there. Do you have a favorite Strava route that you want to tell the world about, or is there socials that you want to promote or anything like that? What do you want people to know?

DENIA: I don't use my socials very much. I do have a LinkedIn.

STEVE MCGHEE: That's totally fine.

DENIA: [INAUDIBLE] But beyond that, I think especially in a world where AI is starting to-- everybody uses AI, from my kids, companies, and obviously in tech-- I think the approach that we're starting to take with applying AI to SRE, or ensuring that SRE is involved in the AI conversation is incredibly important, because it's a way to apply those guardrails that many people are scared of.

A lot of people have hesitation with AI or distrust, and I think the strategies that I talked about today, it's a step on how you can help build that trust on your products, both for your customers but also within your organization. At the end of the day, it's like a step forward to making AI more secure.

STEVE MCGHEE: Yeah, I would say even people who have distrust with just doing stuff to production. Am I doing the right thing? Can I even-- should we try this thing? Having a way to give yourself a little bit more confidence in, like, this comes from data. There's a reason for us to try this direction of travel. I think that is really helpful as well.

MATT SIEGLER: Yeah, thank you very much. You're describing a very measured and cautious and really rational approach, which is really a breath of fresh air.

STEVE MCGHEE: Cool. Thank you, DENIA. Have a good day. Have a good week. Have a good season, I guess. Thanks for coming on the podcast.

DENIA: My pleasure. See you.

MATT SIEGLER: Take care.

DENIA: Bye-bye.

STEVE MCGHEE: OK, bye.
SPEAKER: You've been listening to the *Prodcast,* Google's podcast on site reliability engineering. 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 our 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 Virgi. The *Prodcast* theme is Telebot by Javi Beltran and Jordan Greenberg.
[STATICKY VOICE]