Google Prodcast Season Four Episode 8 | The One with Technical Program Managers and Karanveer Anand

[JAVI BELTRAN, "TELEBOT"]

STEVE MCGHEE: Hi, everyone. Welcome to season four of The Prodcast, Google's podcast about site reliability engineering and production software. I'm your host, Steve McGhee. This season, our theme is 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 remember, hope is not a strategy.

—

STEVE MCGHEE: Hey, everyone, welcome back to the Prodcast, Google's podcast on SRE and production software. I'm joined once again with my friend, Jordan. Jordan, how's it going?

JORDAN GREENBERG: Hello. Good. How are you?

STEVE MCGHEE: What are we going to talk about today, Jordan?

JORDAN GREENBERG: Well, I'm very excited about this episode. As you know, I'm not an SRE, but I'm in technical program management. And today, we have a person who is in from my neck of the woods, I guess.

STEVE MCGHEE: Nice.

JORDAN GREENBERG: Actually, we'd love to have you introduce yourself. Who are you?

KARAN ANAND: Thank you, first of all, Jordan and Steve, for having me here. So my name is Karanveer Anand, and I begin my career as a production SRE in Nutanix. Nutanix is a converged storage infrastructure company. And it was an exciting experience for the production SRE on maintaining the uptime of production services.

Over the time, I've realized my impact is across the one service, which is very narrow impact, and I wanted to broaden my impact to have a broader impact across the different services and tooling. I've transitioned my career from production SRE to technical program management in the SRE field. And then I've grown in my career and then joined Google later on in the cycle, and really enjoying my work here.

Technical program management is more like a force multiplier. And we enjoy the projects that can vary from one field to another, the wide array of projects. And I'm in the Workspace AI SRE team here.

STEVE MCGHEE: Cool.

JORDAN GREENBERG: Awesome. So this is very good. Let's not spoil it. Let's not get too far ahead just yet. I had an interesting way of getting into SRE as a TPM. I first jumped to the TPM ladder, coming from an individual contributor as a support person.

I stayed on a TPM team for a little while and then moved into SRE a little bit later, and the production SRE team, sitting between the services that keep the lights on, monitoring, capacity management, incident response, et cetera. What is your background, the ultimate skills that got you into technical program management, and how did they get you where you are today?

KARAN ANAND: I guess that's a great question. My story is a little different because I always started as an SRE and always had a technical background of the SRE and learned the troubleshooting, how to troubleshoot the services, and read the logs in the early days of my career.

But I guess if you have to put a skill set for the TPMs in SRE, what it's really needed is to get this stuff done. It's very important. But the skill set varies across the career level. It's from staff TPM, it's totally different from senior TPM, it is different to have a charter of portfolio of projects. But as a beginner in this career, I would say getting the stuff done, is a basic attribute which we look for.

STEVE MCGHEE: Nice. So I managed some SRE teams. And part of that is I did manage some TPMs for a little while. And I actually had trouble understanding how to balance TPM work with the rest of the team. And at one point, we actually, within Google-- this is different in every company--

But we ended up moving the TPMs into their own like team of TPMs where they could learn from each other, and they could all speak the same language and everything. And then those folks would be-- they would work with us every day on our team, on some product team or whatever. Is that still the case? Have you seen that model work well/ is that weird? Or do you think that's a good idea for other folks to do, too?

KARAN ANAND: In my opinion, both have their own pros and cons. The one model you are describing is called PMO in some project management or program management organizations where you report to program management, then the program manager reports to the director of program management. So it's like a hierarchy of the whole PMO.

But we have both models here. But I've seen at least in Google in the SRE model, we are more into the

reporting in the engineering chain than PMO, because the context is more important. When you go into the PMO, you lose the context of a project. Because when you're reporting to an engineering director or engineering manager or functional lead, whoever, you are into that context. So basically, contextual TPM is more powerful in these models.

STEVE MCGHEE: So I understand what TPMs do, and I've seen them in a variety of roles. So both of you have worked as TPMs within SRE. Is there anything that's specific about SRE that demands a different angle, a different shape of TPM or certain requirements? What makes an SRE TPM different from everything else TPM, or is that even a valid question?

JORDAN GREENBERG: Let me try answering this from my experience. I'm currently still in GCP. And the sort of things that are important for SRE TPM are like what an SRE would consider but from the more technical side, we have to translate that into the business version of that thing.

So what does it matter if the latency is too long? What does it mean when something is not available at [? 10-9s ?] for the business component. We have to be able to translate SRE concepts into business. Like, if we don't have this, what does it mean for us? Does that match your experience?

STEVE MCGHEE: Yeah, does that make sense to you?

KARAN ANAND: Yeah, it makes sense totally. I guess this is the one big difference between SRE and dev-related TPM things, the contexts we have to translate from technical latency, SLOs to business terminologies. Apart from that, I guess also the first job of SRE is to keep the production services up.

So that's the P0 for-- the bread and butter for the SRE job. And then next comes the project management. If we are doing any project, it can have interruptions due to keeping the service lights on or production services up. So the planning is very different in SRE projects versus dev-related projects.

STEVE MCGHEE: Cool. So given that, can you give us an example of when you were able to help shepherd or make sure that a big project actually completed successfully? I know you wrote an article around splitting up failure domains within a system in order to prevent global failures and things like this. Would that be an example of something like this?

KARAN ANAND: Correct. So like we said, to keep the production services up, we have seen in a lot of postmortems-- SRE has a culture of postmortems. So when we run postmortems, we learn from those postmortems. We have seen from a couple of our postmortems on how we can avoid global outages.

We have seen numerous-- every company has numerous outages, and software is supposed to have bugs.

But how can we not have global outages but have regional outages? Or how can we reduce the blast radius for our outages? That was a goal of the project. So we decided to partition our software infrastructure, which was not partitioned originally.

There are tons of benefits for partitioning the software infrastructure. It was a great SRE project because it spanned out across the different teams. And project management typically was to have tracking how much partition we have done, what is the guidance we are giving it?

So the project layout was more like we have a central team who is responsible for running this project, and then project technical program manager runs the project with central team and running it, farm it out to different product teams, and tracking it centrally, how is the partitioning going, while, like Jordan was explaining, by explaining the business things of a partitioning.

Why do we need partitioning? What is the business impact of this having a partitioning? How much can we reduce the outages? Yeah, and we have finished this project successfully across multiple organizations.

STEVE MCGHEE: I like to joke that SRE is often looked at as the safety team within an organization where if you don't see them, that's good. That means the system is working.

JORDAN GREENBERG: Absolutely.

STEVE MCGHEE: But I think TPMs are the same way. So in this project that you are describing, what would have happened in your mind-- just silly creative, what would have happened if there was no TPM involvement? Would it have not ever started? Would it have gone in the wrong direction? Was there not enough people involved? What are some things that you can take credit for? Take as much credit as you'd like.

KARAN ANAND: I guess TPMs help in a different phase in the different projects. So we can say anything-- maybe the project hasn't started because sometimes we start. We kick it off, hey, we need to meet for this project. We need to have a design doc ready. We need to have guidance ready for product teams.

Let it happen. Maybe some engineers have put down the design document or a guidance document for the whole organization. But we have a saying in management, “if you want to get something done and start tracking, if it's not tracking, then it will never get done”.

STEVE MCGHEE: So the tracking itself is helping it actually succeed. It's not just a side effect. Yeah.

KARAN ANAND: Yeah, right. It's not a side effect. And especially at large-scale organizations like Google, it's

very important to track things and understand the dependencies of each product across the other product. And tracking them down at a regular interval is very important to finish things.

STEVE MCGHEE: Yeah.

JORDAN GREENBERG: Yes.

STEVE MCGHEE: Can you think of any specifics that would have not succeeded or something that you caught that you were like, we didn't x, and then you caught it and we were able to proceed? Is there anything that you can tell us about that?

KARAN ANAND: Definitely. Let's talk about this only partitioning project. So our guidance was not a one-fit-solution-all. And we found a lot of instances where our guidance was not applicable. And we have to do a pilot across the different services and then learn from those pilots on how we can then iterate over the guidance.

So if we haven't done the pilot across the different flavors of a microservices, or we can call box services, then we would be in the waters, because if you roll out the guidance without doing enough pilot testing, then it's not going to work. So we have done that pilot testing with a couple of different styles of teams who are running--

Let's say Gmail has been running their infrastructure for more than 20 years. So they have a lot of services that have been running for a long time. So it's good to test having a pilot on Gmail to learn from their experience as well.

JORDAN GREENBERG: In a weird way, being a TPM can be likened to being the code that sits between independent systems, because the TPM is the communicator between different products that are being represented in the overall system that you're working in.

So while there might be an overarching business goal, there are multiple people that represent those different facets of the business. And if you are on the team that's in SRE, there's a team that is developing new features, that then need to be supported by SRE. There's a team that is marketing these features to people externally.

And the TPM is sitting between all of them to be able to say, SRE is saying that this has X amount of supportability, this SLO. Marketing is saying this is what we're offering to the customer. So they have to basically sit and be the connective tissue between all of these different spaces--

STEVE MCGHEE: Gotcha.

JORDAN GREENBERG: --much like some sort of code could be sitting there to link two different systems together.

STEVE MCGHEE: Sometimes we call that glue code.

JORDAN GREENBERG: Or duct tape.

STEVE MCGHEE: The analogy is apt, for sure.

JORDAN GREENBERG: So you're currently on the Workspace AI team. For our listeners, this is formerly known as G Suite, which is like Docs, Slides, the set of tools that lets you be productive at work or writing up your four-year long Pathfinder campaign notes. Do you have any examples of work that you've done in this space that desperately needed TPM intercession?

KARAN ANAND: Definitely. I guess AI is the most evolution and the fastest-growing industry across the whole world as of today.

STEVE MCGHEE: I've heard of it. Yeah, totally.

KARAN ANAND: And even similarly, Google has been releasing their models very fast. And that's the one part of Google. And another part of Google has to adopt those models at a similar speed to keeping our services on the newest AI models. And it's very crucial not just for safety and reliability for our users, but also to have better performance and efficiency.

There could be tons of benefits to keep running the services on the newest and the latest models. So yeah, we have a workspace for AI. It has a ton of features like Gmail, which helps me write a document or send emails in Gmail. So we have tons of services in Workspace.

So I recently completed a project that involved migrating the services across Workspace to our latest supported models and decommissioning the older and unsupported ones. Yeah, so this required significant cross-functional collaboration with each product team to understand their dependencies, address reasons for delayed migration, why can't they move to the latest model.

If a product team can't move to the latest models? Why can't they move? What do they have dependencies on the old models? How can we solve them? Again, tracking them at a central level to move them to the latest models, and yeah, simultaneously working with the resource managers to maintain the high efficiencies throughout the process and then decommission the old ones to save cost for the company.

STEVE MCGHEE: So are you saying that upgrading to the next model was not a matter of just changing a dev's file from version 7 to version 8? It's a little more complicated than that?

KARAN ANAND: It's way more complicated than it sounds like changing the files, because there are tons of dependencies. We have a lot of things.

STEVE MCGHEE: Yeah.

KARAN ANAND: Yeah.

STEVE MCGHEE: Yeah. Depending on a non-deterministic system turns out to be complicated. So yeah, I get it. Yeah, that's a lot.

KARAN ANAND: And we can't just move the models to a newer model. The AI models are totally different. We have to run a test. If the new models are serving the same performance and accuracy for the old type of workload or not. We need to do, again, different types of testing.

STEVE MCGHEE: Yeah, if it's faster and more available but it gets all the questions wrong, do we want it?

JORDAN GREENBERG: Exactly.

KARAN ANAND: Yeah.

JORDAN GREENBERG: Is there a hardware component in this as well, because when you're upgrading to new models, do they need to be supported by newer hardware to be able to maintain the performance that's needed?

KARAN ANAND: Definitely. There's a hardware component. GPUs, chips or TPUs, which we say are really the main part of this, the whole equation, because we need to migrate to the latest models and make sure they're supported by--

We have tons of different types of hardware models, and need to understand which models are supported by what type of workload. And if we would be-- before the AI era, let's say we are migrating to a new server platform or something like that, it would be quite easy. But with the AI models it's a little different.

JORDAN GREENBERG: So what did this migration mean for the engineering team, compared to what this meant for the business? What did you get to say as a measure of success in this way for the engineering side and the business side?

KARAN ANAND: Definitely. So again, this request came from the reliability organization across the whole Workspace organization. The engineering benefits, I can say, is we can reduce the testing time. Before doing this, we have to test on different models, even the newest models and the older models. But after migrating to the latest models, we can only test on the latest models as we are decommissioning the older models.

So basically, all in all, we are reducing the testing time. If I translate into business benefits, then it would be better safety and reliability, more cost-savings, efficiencies. That's another benefit. Yeah, and reducing the testing time means we can go to the market faster.

JORDAN GREENBERG: Nice.

KARAN ANAND: And this is really required in this AI evolution, speed, time, go faster to the market.

JORDAN GREENBERG: That's super important.

STEVE MCGHEE: I want to back up a second. You wrote an article around TPM and SRE. And part of that-- it was less about project management and timelines and things like that. But the bit that jumped out to me was this idea of you found-- there was like, and you called it the buffer number.

The number was 25%. So I wonder if you can explain to our audience like, what was this number and how did you find it? And should they use it also? Is it something that we should-- is it magic secret sauce that we're not allowed to share. Or is it something that you think the community would benefit from? Well, you blogged about it, so it can't be that secret.

KARAN ANAND: No. Definitely, it's not a secret anymore. We have put it in the public. Yeah.

STEVE MCGHEE: Cool.

KARAN ANAND: So basically, the magic number is just for our organization. It can vary from different organizations. And how we came up with this number depends on how much ad hoc work you get, what is the stability of your services? So why do we need this number? I'll come back to the first-- let's say we'll talk with the first.

As per the project management index, project planning is the most important thing in any project. If your project has planned well, then the success of the project is higher, if you haven't planned it. So similarly, in SRE organization we have to plan for the right headcount.

Because let's say you're working on a project with 20 SREs and your services are not stable and being

interrupted and you keep getting pages, then the interrupts are or ad hoc works are way higher than the planned work. Then the planned work takes a hit and you're going to miss the deadline.

STEVE MCGHEE: Yeah, if you're too busy doing other stuff, you're not working on the project, then guess what happens? It doesn't happen, huh?

JORDAN GREENBERG: Exactly.

STEVE MCGHEE: So you called this the-- what did you mean by buffer number? So can you explain that?

KARAN ANAND: Buffer number means when you plan the headcount or plan the resources for any project planning, keep a 25% buffer or cushion of a 25% headcount based on your stability of your services. So this number could go up, could go down, depending on how stable, how often you get a page, how often you get interrupts in between of the planned work. So this buffer number can keep changing.

The benefits of this buffer number could be, you will not have any project delays. And this project delays are very bad. No one wants the delays. Everyone wants to deliver the project on time.

STEVE MCGHEE: Cool. It sort of sounds like how we think about toil with regards to-- you don't want to do toil, but you know it's there. And it always changes all the time. So I know with toil, we recommend, or I recommend at least, that people try to measure it multiple times a year and just see how we're doing.

Because even if you reserve like X amount of time for it, that's not necessarily what's going to really happen. Is this the same deal here? Do you think that it's worth checking back in with your teams to make sure that the buffer is what we think it is or something like that?

KARAN ANAND: It's very important to keep visiting this number and verify because you don't want to underutilize or overutilize the resources as well. So it's very important to keep this number checked and have a healthy balance between why you don't want to have a key high buffer versus why you don't want a low buffer as well. So we perform this activity every quarterly basis to check.

STEVE MCGHEE: Cool.

JORDAN GREENBERG: Wow.

KARAN ANAND: And it's really helping us so far. Yeah.

JORDAN GREENBERG: That's awesome. Speaking of helping, what are some ways we can leverage AI to assist with furthering the SRE and TPM relationship? How do we use our keyboards to help the engineers

use their keyboards?

STEVE MCGHEE: Let me see if I can interpret Jordan's question, because I think I know where you're going with it. But how would you like to use AI yourself as a TPM in SRE to help the teams around you, or to engage better with other teams, or whatever it is? What are some other applications that you yourself would use as a TPM within SRE? Is that right, Jordan?

JORDAN GREENBERG: Yeah. And then, the force multiplier is the TPM, but AI is the force multiplier for the TPM. So it kind of scales it out in this strange way.

STEVE MCGHEE: Multiplier multiplier.

JORDAN GREENBERG: Exactly. Now we've got exponential force multiplication.

KARAN ANAND: Yeah.

JORDAN GREENBERG: So what does this mean for being a TPM in SRE? And how can we ultimately support our engineering teams better with the assistance of AI?

KARAN ANAND: Yep. So AI is a multiplier and TPM is a multiplier, so we are doing exponential multipliers here for AI with the help of AI. And we have already developed a lot of bots inside to do a postmortem analysis. SRE has a culture of postmortem. So we have a postmortem analysis.

Then AI is helping us by giving the top risks. When AI is giving me top risks, it's helping me-- it's funneling my 2026 roadmap already, where TPMs are responsible to create a roadmap for the next year by working collaboratively with engineering leaders. But we can give a great input on what are the top risks based on postmortems of 2024 and 2025, the last couple of years.

STEVE MCGHEE: So is this something that you've done? Has it been working for you?

KARAN ANAND: It has been working. We have a small-- we ran a hackathon in our organization.

STEVE MCGHEE: Cool.

KARAN ANAND: And we came up with this bot. And we have a postmortem analysis. A couple of months back, we had to do the postmortem analysis manually, go through each postmortem, understand the root cause, and how can we mitigate this and all that. Now we have a postmortem written. The script can crawl through all the documents and figure it out.

STEVE MCGHEE: Cool. So when I was running a team-- so I ran Android SRE a long time ago. And I had one TPM for all of Android, which seems like a bad idea. It was pretty wild. In fact, we didn't have a TPM for a while until we got one. So shout out to Ron. Thank you, Ron. You saved our bacon.

JORDAN GREENBERG: Nice, Ron.

STEVE MCGHEE: And then we had more and more and more. People have heard about dirt. So there's been a lot of publications. So there's a woman named Kripa who wrote a bunch of things. I remember her being kind of an early TPM, at least one of the more outspoken, like, would talk outside of Google about TPM stuff and things like that.

So I think TPM has grown inside of SRE quite a bit. So of course, this is a leading question, which is like, what's next? How do you think TPM inside of SRE, or maybe alongside SRE, whether it's inside of Google or outside, how do you think it's going to evolve over the next two years? Like, it's going to grow. We're going to use more AI, something like that. But what can you say that's more interesting than what I just said?

KARAN ANAND: Yeah, it will definitely grow. With the evolution of AI, we need more TPMs in the AI for the SRE side as well. We need to keep-- the trust and safety part for the AI is very important, even for the user. Although AI is bringing the productivity high for billions of users across the world, it's very important to keep this trust and safety high.

That's where the reliability comes in. The reliability has an overlap of trust and safety. So the TPMs will keep growing. It's just we need to know how to use the AI and when to use the AI to keep the productivity high. But I don't think the TPM's job with the evolution of AI will go away. TPMs will just become more productive, but it will not go away.

STEVE MCGHEE: Yeah, the work keeps growing, I think. Yeah.

KARAN ANAND: And we keep getting new work and work. And especially the speed has increased a lot in the last few years.

JORDAN GREENBERG: Yes. It's definitely gotten a lot easier to do these types of analysis and to see it in a way that's very SRE-coded, is awesome, specifically with postmortem analysis. That makes a huge difference in the amount of time we spend. Thank you very much, Karanveer, for taking the time with us today to talk about SRE, TMPery.

STEVE MCGHEE: Thank you, Jordan, for coming on as well. This has been a great conversation. It's been really good to actually go back to my roots and remember what it's like to work with a big team and solve

seemingly intractable problems than having someone come in and be like, listen, we can figure this out. You don't need to just YOLO the whole way.

JORDAN GREENBERG: Yes, that's true.

STEVE MCGHEE: We can actually come up with a plan. It's like, all right, good idea.

JORDAN GREENBERG: Yeah. And lean on your TPMs, I'll say that as a closing argument. We're here to help you, and we want to be the person who does whatever they can to make sure that you're able to stay on the keyboard and do what you're really good at.

STEVE MCGHEE: Last thing is, I want to get, Karan, on the record, do you think that we should waterfall plan all of our projects in SRE? You think that's a good idea, like just come up with the perfect plan up front and then just never touch it ever again?

JORDAN GREENBERG: That works a lot.

KARAN ANAND: I don't think so.

STEVE MCGHEE: No, no, no, no. Yeah, yeah.

KARAN ANAND: Especially with the AI things, we need to keep it agile and make sure teams are accountable and running faster. Yeah.

STEVE MCGHEE: Yeah. Yeah, it's an allure, I think, that a lot of people suffer from when they're like, listen, I have the best plan. Let's just do this, and it's going to be great. And then I think a lot of times TPMs come in and go, are you sure? And then check in every two weeks like, are you still sure? What has changed? Let's reassess.

So I think that's great. Having folks that can keep everybody else on track with regards to that and get it like the whole time and be like, look, I wrote a script and I'm afraid you're wrong, whatever your assumptions were. That is fantastic. So hats off to you. Thanks very much. It's been great having you on the podcast.

KARAN ANAND: Same here.

STEVE MCGHEE: And thanks, Jordan, for joining us again. And I guess we'll see you all next time. Oh, last thing, Karan, is there anything that you want people to connect with you on your socials or anything like that?

KARAN ANAND: People can connect with me on my LinkedIn. Karanveer is my handle. Yeah. Feel free to send me requests.

JORDAN GREENBERG: Throw that in the end, too.

STEVE MCGHEE: And you have a couple of blog posts we'll link in the description. Those are great. I've read them. They're awesome. And yeah, keep it up. Keep making Workspace smart.

KARAN ANAND: Yeah.

JORDAN GREENBERG: Thanks for joining us.

KARAN ANAND: Thank you to both of you for having me. It was great talking with both of you.

JORDAN GREENBERG: Thank you. Bye.

STEVE MCGHEE: Bye.

KARAN ANAND: Bye bye.

—

[JAVI BELTRAN, "TELEBOT"]

JORDAN GREENBERG: You've been listening to Prodcast, Google's podcast on site reliability engineering. Visit us on the web at SRE dot Google, where you can find papers, workshops, videos, and more about SRE.

This season's host is Steve McGhee, with contributions from Jordan Greenberg and Florian Rathgeber. The podcast is produced by Paul Guglielmino, Sunny Hsiao, and Salim Virji. The Prodcast theme is Telebot, by Javi Beltran. Special thanks to MP English and Jenn Petoff.