Google Prodcast Season Four Episode 3 | The One with AI and Todd Underwood

[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 Podcast. This is Google's podcast about SRE in production software. Today, we have a special guest, tmu. Everyone knows what that means, tmu. In real life, his name is probably Todd Underwood. Maybe that's what he writes on actual forums and things.

And of course, I'm always here with my friend, Matt. Hey, Matt. How's it going?

MATT SIEGLER: Hi. Good to see you again.

STEVE MCGHEE: And I think the two of you, actually, Matt and Todd, might even be in the same part of the world. Is that even true? Is that possible?

MATT SIEGLER: That's true.

TODD UNDERWOOD: We're like, fewer than two kilometers apart.

STEVE MCGHEE: Whoa. Crazy.

MATT SIEGLER: I'm, in fact, at the moment, in his old office.

STEVE MCGHEE: Oh, even better.

TODD UNDERWOOD: It's a weird experience. It's a little bit--

STEVE MCGHEE: Surreal.

TODD UNDERWOOD: --insulting, frustrating, unnerving. I don't like it that you've taken over. It's fine. It's fine. I'll get over it.

STEVE MCGHEE: It's fine. Yeah.

MATT SIEGLER: I mean, I'm in his old office. I'm not at his old office, in the sense that I'm borrowing his old office.

TODD UNDERWOOD: I can't tell that. I mean, I just see you occupying it. See?

MATT SIEGLER: That is the correct word. I'm occupying his old office.

STEVE MCGHEE: Todd, who the heck are you? What's your deal? Can you tell our listeners?

TODD UNDERWOOD: I am Todd. I lead reliability for Anthropic, which is an AI company, working to develop safe, beneficial, and harmless AI. Prior to that, I worked at OpenAI for a brief period and at Google for a very long period. So I have been doing reliability stuff for most of my career, but I've been doing Machine Learning reliability stuff in particular since about 2009. So yeah, that's who I am.

One piece of advice I have for everyone-- do not write a book. Under no circumstances should you ever write a book. I know this because I wrote a book with some folks, about Machine Learning reliability. It was a catastrophic mistake. Really terrible. No, the book is fine, but it's really so much more work than you think it is.

STEVE MCGHEE: I wrote a booklet one time, and that was hard. It's basically just a Google Doc, but very long.

TODD UNDERWOOD: Yeah, like a little booklet. Oh, man.

STEVE MCGHEE: It takes a long time.

TODD UNDERWOOD: The worst book story I have is, you get to the end, and you print it out for the first time, and you're actually reading through. So you read through the whole thing in order. And my teenager came down and saw me, and said, what's that? And I'm like, well, this is the book. It's almost done.

And she said, like, wait, you were serious about that? I've been working on it for like two years. She was like, oh, everybody's writing a book. Everybody's got a screenplay. I'm like, no, we were writing a book. We wrote a book.

STEVE MCGHEE: This is different. It's real.

MATT SIEGLER: Well, tell us about the book. What was-- it was an important book. Tell me, what happened after the book?

TODD UNDERWOOD: I think what's interesting about-- so the topic is tough because when you think about what we do, what do you all talk to people about making things reliable? Oh, my God, look at that. You've got one.

MATT SIEGLER: Yes, reliable Machine Learning.

TODD UNDERWOOD: And also, what's creepy about that book is, Anthropic employees are called ants. I was like, this is weird.

MATT SIEGLER: On the cover is a bunch of ants. Yes.

TODD UNDERWOOD: This is very strange. It was destined to be. What was interestingly difficult about the book is, we intentionally tried to give it a little bit longer shelf life than you normally do for technical books, in part, because we just thought like, this stuff changes so fast. If you're going to try to address what people are doing today, don't write a book, just put out a blog, write some web articles about it, do a video tutorial or something.

But we wanted to try to get at some of the underlying principles, things like, how do you think about managing data and metadata? Or how do you think about troubleshooting incidents for services that have AI, ML as a particularly important component of those interfaces and applications? And so to do that, you have to be a lot more generic and a lot more principled, which is hard, because you want to be concrete enough to be useful but not so concrete as to be useless in two weeks.

I think that was what was hard. I think the reception was pretty good. But also, what's weird is it's been super steady. I would say, the number of times people have brought up the book in the last month is actually not dissimilar from this month last year, which is really surprising for a technical book.

STEVE MCGHEE: That's cool.

TODD UNDERWOOD: Yeah.

STEVE MCGHEE: In very recent news, I wanted to tell you that yesterday, I spent much of the day, vibe coding for the first time in my life. And it is a phrase, so this phrase may not last in perpetuity as people discover this podcast 25 years from now. But--

TODD UNDERWOOD: Indeed.

STEVE MCGHEE: --I had an AI write code for me, and I just accepted whatever it said. I was like, yup, yup, yup, yup, yup, yup, yup. I didn't do any debugging. And all I did was, I write tests and now make the test pass. No, not like that. That was the level of my prompt engineering. It wasn't very smart. It was just like, hey, man, come on. And it worked. It was pretty cool.

Some of which was like production-ey things-- it wasn't real production, of course, but it was like, make this so I can run it on a cloud, and like, make it so that it is debuggable and add logging, and like, blah, blah, blah, blah, blah, words like that. So this is a ruse to get us to talk about using AI in production, because that's the real topic of today.

TODD UNDERWOOD: Yeah.

STEVE MCGHEE: So how do people actually use ML AI in their day job if they're SREs, if they're people who listen to this podcast today? In your opinion, so far, today, as you've seen it, actually be good.

TODD UNDERWOOD: Do you mean like, what is the current state of affairs?

STEVE MCGHEE: Yeah. Current. Today. Yeah.

TODD UNDERWOOD: So that's tough. So let's think about from minus five years until plus 100 years from this moment. So within the minus five years to this moment, we have all been on the receiving end of this huge amount of hype of like, oh, AI is going to do everything for you. This goes back to a bunch of AI startups that were doing a thing that I haven't heard people use the term as much recently, but AIOps, which is the idea of using AI in production. That was around.

STEVE MCGHEE: That was still around, for sure.

TODD UNDERWOOD: OK. So that's still around. You would know better than I would. So people started trying to sell that for anomaly detection, for automatic configuration management, for dashboard creation and curation, for metric selection, for all this stuff, for config authoring.

And let's be honest, like most of it hasn't worked very well. It does really well on demos, but then you go to use it on a real code base, and it's like, I don't really know what's happening here. Maybe you could do a bunch of work to make your code base, just like what I was expecting. And then I will save you time. And it's a trap.

So that's where we have been until now. 100 years from now, I think we all agree that computers are going to do most of this for themselves and for us. We're not going to need them. So somewhere between the moment that we're at and 100 years from now, things are going to change a lot.

So I've seen some limited stuff that's starting to work, and it's mostly stuff that's super adjacent to software engineering, as you say, Steve. So can you write Terraform for me? Yeah, that stuff can write Terraform. Can you produce a Helm chart? Yes. Can you help me troubleshoot this particular Kubernetes command that's not working with-- yes, they can do that.

But then later, when you say, hey, could you specify and architect a series of redundant services spread across three locations with 20,000 nodes in one location and 2,000 nodes in each of two other locations, and describe that, which is super normal for a human. You're like, hey. And they would think about it. No, they're not there yet. So somewhere between, do some Kubernetes for me and build the service, that's the gap that we're in.

STEVE MCGHEE: Make it so it's good? Yeah, that's hard.

TODD UNDERWOOD: Make it so it's good. Well, in thinking about tests, this is the thing I have to say, a little side note. This is the thing I find most human about the coding. If you tell an AI system to make something pass the test, the most common thing it will often do is change the test to be passing.

And I'll be honest with you, a lot of us feel like that as well. And a lot of software engineers do that too. You're like, what is the fastest way to get this code to pass those tests? The answer is, have the test return true. That's the fastest way.

And lots of us do that. And honestly, if you told a very junior developer who wasn't really familiar with test-driven development, they would do it too, because they're like, this thing says no, but if it said yes, then the next step would happen. Make it say yes. Oh, look, I can do that. It's even less code.

MATT SIEGLER: All right. You took a wide spectrum of discussion through like where AI applicability from down to the ultra-tactical working test munging, and code, and up at the super strategic, doing big plans, doing forecasting.

Where are you seeing actual leverage happening now? Where are people finding, oh, this is really useful. I'm able to do this. I'm not toiling because this machine is doing the work, so I can do some more of the thinking. Where's this happening now? Not 100 years from now, not five years ago, where it was garbage. Where are we actually seeing some progress that's safe, that the risks are worthwhile, that everyday, people can actually do things?

TODD UNDERWOOD: I think we're still in the very early stages of that. So we're in the messy middle, but we're at the beginning of the messy middle. And what I mean by that is, before we had these kinds of computer systems, people like us, we wrote the config for ourselves, and people like us, as coworkers, reviewed each other's configs. That's how we did it back in the dawn of time. And then--

STEVE MCGHEE: Last year, yeah.

TODD UNDERWOOD: Yeah. Last year. Conf T! But at some point in the distant future, humans will not be involved at all. And that will be a very different world, where the computers talk to the computers and the representations of what they do, and also, what they need to be good at, is very different.

In the middle, the computers need to be able to help us. And that's where we are right now. So what I see right now are very beginning stages of people using computers to help them with production, not very sophisticated.

Cases that are incredibly controversial, and don't work at all, and might never work, or will not work with our current approaches, are anomaly detection. That's the thing most of us want is anomaly detection. I don't think we're going to get that anytime soon.

STEVE MCGHEE: Nontrivial anomaly detection.

TODD UNDERWOOD: Yeah, I mean--

STEVE MCGHEE: We already get the basic, like, ‘eh well, maybe’, kind of stuff.

TODD UNDERWOOD: That's right. But then those are either huge false positives or huge false negatives. They're like, oh, that thing spiked. Well, it spiked because it's the beginning of the working day.

STEVE MCGHEE: Yeah, it's important to point out. There already is anomaly detection, it just sucks. It's in many products today. It just sucks.

TODD UNDERWOOD: Yeah. Well, because Matt did say, reliable.

MATT SIEGLER: Yes. Yes. Yes.

STEVE MCGHEE: Yeah, fair. OK.

TODD UNDERWOOD: So I don't think the anomaly detection stuff is very useful. I do think there's a couple use cases that I've seen that are pretty good. Steve Ross and I wrote a paper a number of years ago that pointed out that the cases where people are using these systems, where it works, we don't even think about. So let's talk about autoscaling with your current employer.

Autoscaling has been ML-driven at Google for a long time. And you don't even think about it. And you're like, well, yeah, that's right. It is just because they had some simple model of a low watermark, bump things up, high watermark, bump things down.

And then they were like, oh, well, what if we do some diurnal prediction? What if we store a time series? And what if I say like, hey, I think, in 26 minutes, you need twice as many instances of your job spun up? And they just get ready to do that because spinning up instances is not instantaneous. That was a good little pun to that.

STEVE MCGHEE: Got it.

TODD UNDERWOOD: Yeah. So things like that already exist. In terms of the work that people do out in the real world, the only stuff I've seen that is pretty much working is things related to first draft configs and first draft designs.

And also, there's a little bit like, I got paged, what do you think I should look at? Those things, where there is something that is useful but is not left to its own device. It's going through a human. So in that case, Matt, I alert you, you're on call for some service, and you get an alert.

And if you're not great at the service or you're not super experienced, which happens to us all from time to time, it's pretty great to go into a page and say like, hey, I got a page and have the model say, like, I'm not really too sure, but I think you should look at one of these five graphs. I've got five graphs. I think you should look at those.

And if three of them are useless and unrelated to what you want to look at, but two of them really help you understand what's happening and how to fix it, that's pretty great.

STEVE MCGHEE: Yeah.

TODD UNDERWOOD: The last thing that I've seen is super boring. I talked about this last year, but what's actually really interesting is the replacement of interfaces to technical documentation. OpenAI had this where, instead of saying like, let me search a set of web pages, you instead interact with an AI system that has access to the full set of technical documentation, which means you can explain, expand, and correlate.

You can say, I don't understand how this system works and say, well, this is a system that does that, and it relies on that. So the interface to that is much, much better. Tell me all the things that control access to files stored in my blob store.

That's not an easy question to ask of your documentation. That's not a web search kind of a question. That's an AI system, kind of a question. But if you have a huge amount of documentation, and you ask it, most of the good AI systems will answer that question for you.

STEVE MCGHEE: So this sounds a lot like toil reduction through this type of technology, as well as freshness of understanding. I don't know. There's probably a better phrase for this, but being able to know less about what was written down the last time we wrote things down and more about the state of the system via many sources, potentially.

TODD UNDERWOOD: I think that. And I think that in the human augmentation piece of this, all of the systems, where you can put a series of documents into a project and look through that project, so like NotebookLM at Google, Anthropic projects, both of these are like, hey, link to these documents, either by uploading the documents, or linking to your Google Drive, or whatever. And like, now, I just want to talk to the documents, I want to ask questions about them.

And so cool examples I've seen. I mean, this is going to be boring from a production engineering point of view, but I'm a manager. And one cool example I heard was, put all your one on one docs into a project. And then you can ask questions about your one on one docs in aggregate. You're like, who is really struggling? Or who has had some interesting questions? Or who have I met with least in the last three months?

You could get these things from other places, but you can put your team doc meetings in there or your incident review docs in there and say, what were the most contentious incident reviews in the last six months? And you're like, you can figure that out, but being able to zone in on like, hey, here's three that had a large number of comments, a large number of discussion on the comments, failure to resolve on what the follow up items were.

All of these are in this bucket of helping humans be more useful, human augmentation. None of these are the, oh, and then the computers just go away and do it right. We're not at that point yet.

MATT SIEGLER: This really brings something that is top to mind for some people is trust and safety is knowing that what's coming out of the other side of these decisions is not missing what's important but also not suggesting something that's potentially problematic. What's in your head about this right now?

TODD UNDERWOOD: I think, the main thing that I've been enjoying with some of these systems that are based on either web documents or stored documents in a product are citations. So don't tell me like, this is a thing. Tell me why you think that's the thing, so that I can build trust.

That works with humans as well. Somebody tells me, hey, the weather's going to be terrible tomorrow. I'm like, I didn't know that. Why is the weather going-- why do you think that? That's perfectly normal. Where did you get that information? Because I want to have access to that information and understand it better.

That's a normal thing to do to another human being. It's also a completely normal thing as you're trying to build trust in an AI. Don't give me a number. I mean, sure, give me a number, that's fine, or give me a set of numbers, but also, tell me, where did you get that information so that--

STEVE MCGHEE: Citation needed.

TODD UNDERWOOD: Yeah.

STEVE MCGHEE: For sure.

TODD UNDERWOOD: Citation needed. And the systems like Anthropic projects or NotebookLM where you're working with a series of documents, having deep links into the document. They're like, look at this table, or look at this paragraph, or look at these three things. That's why I think that this is important, or this is wrong, or this is interesting, or this trend exists.

MATT SIEGLER: I like that answer. Also, at Google, clearly, we've been trying to get people to dig into, not just here's the answer, also go look at where we got it for you with authoritative information. I think what you just said was, this is a citation. This is where we think that came from. How do we get people to want those kinds of answers, not just, here's a summary of the answer?

This is the credibility of that answer. Go look and read for yourself. Go make a judgment, not just take this answer as it is, because I see people just taking the output of that, digesting and going oh, that's the answer I'm ready to go, but I don't think we want people doing that. How do you suggest we have a literate way of interacting with these machines?

TODD UNDERWOOD: I mean, I'm not an expert in-- worse than that, long ago, I discovered, I don't really understand other people all that well. So that's fine. So I don't know how you get anyone to do anything. What I do think is that when I think about the way I use, for example, search, I think there's some different use cases.

One of the use cases is like, is Cedar Rapids the capital of Minnesota? No. No, it's not. OK. What is the capital of Minnesota? Tell me about St. Paul. Fine. OK. So I just wanted to know a fact.

The other question is like, how are capitals of US states selected? Or why was the border between Canada and the US picked to be 45th parallel? Why did that make sense? That's not a one fact answer. I want to do some reading, but I also want to look at citations about the reading.

So I think, as we move away from just simple answers into more complex stuff, I do think more and more people will be interested in the summaries and the links about more of the summaries. But I think there is a UX issue there, where early versions of Perplexity, for example, and some of the other search engines didn't really-- some of the other AI search engines, I mean, sorry, did not really highlight the sources that well. And that's not great, right? Because you really want to be like, hey, I think this, link, link, link. I think this, link, link, summary from these links, and then also have a table of citations at the bottom. That's a weird UX, but I think those are the important things.

STEVE MCGHEE: So just focusing on your position today, so you’re the reliability czar will say, no, you probably have a real title. You're a reliability dude at a place.

TODD UNDERWOOD: Yes.

STEVE MCGHEE: Does quality fall under your purview? Do you care about this stuff? And if so, this is where hallucinations come in. So someone asks about the land dispute between these two borders.

A hallucination can totally slip in there. Do we know? Do you know? Does wearing your reliability hat at a company that does this stuff? Is there a way to get a graph to be like, wasn't great?

Someone said this was wrong, or we think maybe, the model was being weird. Does it just happen? And we're like, I don't know, tough. Or is there a way to make this better over time? How do you come across this type of quality issue? Like it's up, but it's weird.

TODD UNDERWOOD: So it's funny. You stepped into my trap, Steve. I've given a number of public--

STEVE MCGHEE: I thought you stepped into my trap, but here we are at an impasse.

TODD UNDERWOOD: I've given a number of presentations arguing that end to end model quality is the only SLO that people working on reliability for ML systems can have. And I think it's easier, hallucination kind of stuff is interesting, and some of the subtle cases are interesting. But let's just talk about a payment fraud system.

If the payment fraud system is running, and the model is fresh, and the model just says, yes, every transaction is fraud, the system is not running. You are no longer accepting any payments. It's not an interesting case. The model just thinks that everything's a fraud. OK. Is that OK?

STEVE MCGHEE: Might as well be down.

TODD UNDERWOOD: Right? Might as well be down. Right. And so I think, similarly, if you're Amazon, and the recommendation systems recommends a kitty litter robot to everyone, regardless of what they're purchasing, Amazon's losing tens or hundreds of millions of dollars until they fix that recommendation, because, I'm not saying no one would buy it, but I don't have a cat, so I'm not buying a kitty litter robot.

And so that's the first thing I would say. And then the second thing is, even in the modern systems, one of the most common intersections or one of the most obvious intersections between model behavior and reliability is like the trust and safety checking.

Like at Anthropic, we care desperately that the models are safe, that the models are helpful, and that the models don't let you do bad things. We published this whole responsible scaling policy about what we're going to prevent the models from helping people to do. We don't want the models to help people carry out acts of terrorism.

So for example, you want a model that understands chemistry and biology, but you don't want a model that helps you make bioweapons that will kill people. So that's tough, because some of that's just chemistry and biology. So that's a tough line.

So even after you test the models, the nefarious people, sometimes, researchers are trying to get the models to produce harmful content. And there's some live checking that goes on. If that live checking is bad, it can just disable all of the sessions, be like, nope, nope, nope, nope.

STEVE MCGHEE: You still want people to be able to make real fertilizer, not fertilizer for bad things. Yeah.

TODD UNDERWOOD: That's right. That's right. Yeah. For example, it's completely reasonable for somebody who's got a farm to be like, I am sick of paying for-- is there anything I can do to not have to pay for this stuff? Yes. But also, yes, there are bad uses for similar chemical compounds.

And so I think that's what's interesting is that-- so what I would say is that there's a tendency for SREs who work on the stuff to be like model quality, not my problem. Well, literally, the only reason you're running this system is because the model does something. If the model didn't do something, you wouldn't be asking it questions in the middle of whatever you're doing, whether it's targeting ads, fixing spelling, correcting grammar, having a conversation, identifying fraud.

The model does that thing. So given that it does that thing, if it stops doing that thing, then you don't actually have a service anymore. It's a complex issue, though. And I find it interesting, because obviously, if I'm running a model, I didn't make the model. But I think there's some good patterns on how to do that.

So one of the simplest patterns is, if the model is brand new, and nobody's ever used it, and it's terrible, and Matt, you launched it, you launched a bad model, you take it back. It's all your fault. I'm not going to help you.

If the model's been working fine for two weeks, four weeks, and nobody's touched it, and all of a sudden, it's garbage but so are five other models, that's probably my fault, not Matt's fault. It's probably not that all five models went bad at once. So you can do some really simple correlation to try to figure out, is this an ML problem that is particular to the design of the model, of the training of the model, or is this some kind of a systems problem?

STEVE MCGHEE: Yeah. Life is systems, man, for sure.

MATT SIEGLER: This makes me think about a core principle, which is an SRE principle of making conscious decisions between product velocity and product reliability. And I want to hear your thoughts on where you see industry-wide AI product releases coming in terms of reliability versus velocity right now, because these releases are coming hard and fast. And they're coming hard and fast because the discoveries are coming hard and fast, and the models are coming out really quickly.

What is your stance? What are you seeing right now? Where are we in this pendulum between things coming out faster and things coming out better? And what are we to do?

TODD UNDERWOOD: I think it's been an interesting question. What's funny is, that question, I dealt with when I was working on the Google Cloud AI team back in, I don't know, 2022, or 2021, or something. This is a long-term question, long-term for our industry.

STEVE MCGHEE: Ones of years ago. Yes.

TODD UNDERWOOD: [CHUCKLES] Ones of years ago. Yes. But I think that the right way to reframe this, the way we often do in reliability circles, let's think about the end users and their preferences and just say like, hey, I got a new thing, or I can spend time making the existing thing better. What do you want?

And right now, the market says, hey, I just want the new thing. I just want the new thing, mostly. I think, there are some cases where that's not the case, but the feedback I've heard from most users is, similarly, you can often trade reliability for capacity, as well as you can trade reliability for velocity. You can be like, well, I can run the service hotter, but it's going to break more often. That's another common thing.

And most users will be like, yeah, yeah, I'll take that. I'll take it. I would like twice as much quota at 1 and 1/2 or 2 fewer nines of reliability. I'm like, really? Right now, a lot of users are saying yes to that. I think that the key question, the same way that cloud was like a toy until it wasn't, cloud was just kind of a thing people played with and put some stuff in, until you turned around and looked, and 20% or 30% of all the 911 services in the country were running on somebody's cloud.

You're like, oh, OK, so not a toy, need to be very thoughtful about how we do this stuff. I think we're going to get to the same thing with the public AI systems. But I still don't think we are. I think users want a better model.

STEVE MCGHEE: If we're doing this really quick, because this is what the market is asking for, and this is what we can deliver, and like, OK, sure, market, you say so. What if we're wrong? What's the backup plan? What if it goes too fast?

The worry that I have, and this is pretty philosophical, is that this is a highly-leverageable technology. We can get it to the hands of literally billions of people. And if it's goofy, or weird, or wrong, it can have drastic effects real fast. And it can have a lot of them, potentially, especially in the, not this year but next several years mode, presuming that we're right about that.

Are there thoughts about that that you're aware of within the industry, within things that you work on in terms of, if it's bad, what do we do? Is there a roll it back big red button in the meta sense, I guess.

TODD UNDERWOOD: I mean, I think, when Matt was talking about this, I was thinking about reliability in the, like, the model's fine, but it's not working right. The inferences are not returning. I think, in terms of launching new models, one of the things that Anthropic is particularly opinionated about is being very careful and thoughtful of that.

So I think it's a great framework, this responsible scaling policy. And the responsible scaling policy just says, if you have a model that can do these things, then you need security controls that can do these things. It's very explicit.

And before I started at Anthropic, I thought like, oh, this is the kind of thing that security orgs do to market themselves. When I got to Anthropic, I asked Jason Clinton, who is the chief information security officer, I was like, hey, how does security work here? He's like, well, I mean, we describe it publicly, and the responsible scaling policy, that's just what we do.

Every time we have a model, we decide how capable the model is, and then we implement the controls that are of that capability. And so that's what we have now. And we're trying to encourage other organizations to do similar sorts of thinking. Because I think you're right, the ways that humans use these things are many and varied. It's exciting, it's wonderful, but the potential for harm is pretty significant as well. I don't know if you all have read all these stories and talked to friends who are like, oh, yeah, I use ChatGPT as a therapist. Really? The models, they're not suited to that, not that they couldn't be. I can imagine a model being a fantastic therapist, but I'm not aware of any data that says that we have models that are good at that and safe to do that right now, because that's tricky.

So what I would say is like, yeah, everyone-- like, Anthropic is super careful about model releases for these reasons. And we're hoping to pressure the rest of the industry into being careful about it as well.

STEVE MCGHEE: In my experience, and I think yours, because we worked together at the same place for a long time, like SRE, historically, or at least, yeah, I would say, SRE, in general, has been at this weird intersection, where we're like the ‘fight for the user’ group, where we would say like, no, we shouldn't do the thing. Even though some team was like, this feature is going to be great, we would hold this sign saying like, no, it's actually not good. Do you think that SRE in the future, the current, is in a similar position to be able to say like, yeah, but not cool? This isn't going to work for the following reasons. Or is it like, are we still in that position, where we're looking at the holistic picture, where maybe, other people at the company are not?

And is this a bad idea to try to do this? Or is this actually an honorable thing to try to do at these types of companies?

TODD UNDERWOOD: And the origin behind that was always going back to early days sysadmins, who are like, I have access to all the secret data. And I take that responsibility super seriously. A lot of sysadmins I know are as ethical as the physicians I know about private data. Of course, I know what you do on your computer. And of course, I would never let anybody know that without appropriate permission, et cetera, et cetera.

I think that the difference here is that the worrisome capabilities are much more subtle. They're not like, I'm going to expose your credit card information online, or I'm going to give somebody access to your bank account. They're, I'm going to launch a model that is weird and dangerous for a very small subset of people under this very odd circumstance.

And so I guess, what I'm getting to is, one of the things I really like about my current employer is that the whole company is organized around trying to prevent this. And I think it needs that, because you need sophisticated researchers. You need a trust and safety team that understands those. You need red teaming. You need people publishing about that stuff.

There was one of my coworkers, who, until recently, was here in Pittsburgh, did a whole paper of like, hey, I used one of our models to do end to end network penetration. It worked. And so I figured out how it worked. And it was really easy and easy to do on everybody else's models too. And I was like, can you publish that? He's like, yeah, it was so easy. I'm sure other people are already doing it. We publish it so that people can see how this is done.

So this is my point is, that took a PhD researcher four weeks, which is a small amount of time, but also not something an SRE is going to do on a Tuesday before their call shift starts at noon. So it takes more work.

MATT SIEGLER: Tell us a story, an anecdote, something that really surprised you, something either adjacent or outside your field since changing companies, you meet new people, new scenarios, something you've learned since changing that really took you by surprise.

TODD UNDERWOOD: I will say, here, speaking directly to us, as the audience, and what I believe is your audience. Most of us we're pretty skeptical about these technologies. We're like, oh, they're OK. They fail a lot. I don't know. I don't really trust them. I'm not sure.

Also, AI is just a bubble. Also, these are just glorified zip. It's just some glorified compression. LLMs are not-- I'm like, OK, I think those are widely-held beliefs. And when I was at OpenAI, it was difficult to disprove those because OpenAI is such a consumer-oriented company.

Consumers are very susceptible. We love us a fad. We just love whatever other people are doing, especially like nerd land, where if three other technical people are doing a thing, I just have to do that thing.

STEVE MCGHEE: I'll take two. Yeah.

TODD UNDERWOOD: Take two. Right. Then I came to my current employer, who really sells mostly to businesses. And it's not that businesses are not susceptible to fads, but all of them are doing proof of concepts-- proofs of concept. That's better. And they're buying a lot.

And so I think, what surprised me is how much of a real market for these services there is and how diverse it is. Clearly, the coding use case has caught wildfire. People love this for coding. It's very concrete. It's super economically valuable.

The price points are right. It's going to change our whole industry. And we can all already see it. That one has appeared, basically, out of nowhere in the last two years, where, when GitHub launched Copilot, I think a lot of us were like, cute story, but it's not that useful, in part, because the models weren't quite good enough yet. It was a good UI, but the models weren't good enough. But now, Steve, you're vibe coding, and it's working, that's wild, but also for marketing.

And also, for marketing, like product marketing, someone was describing to me how one of their products is, somebody writes a strategic product brief and outcomes, the marketing segmentation plan, the advertising plan, the manufacturing plan, what ingredients they should or should not use, proposed names, blah, blah, blah. And I'm like, oh, is that good?

And they're like, yeah, we have currently between 112 and 120 people doing that to each product. And if you can do that, those people could go do something else. And it doesn't do it all by itself. But anyway, my point is, the thing that I have seen that has surprised me the most is the volume, the intensity, and the diversity of real, economically valuable use cases that already exist in these models that are far from where they're going to be in the near future, so that stuff.

STEVE MCGHEE: Nice.

TODD UNDERWOOD: I think, the reasoning stuff, the little subpoint, the reasoning technology that has appeared recently is fascinating. I don't know if either of you watched ClaudePlaysPokemon. That is mesmerizing. It is mesmerizing.

So basically, they just turn Claude with reasoning turned on at Pokemon, and they say go. And it gives it an interface and so it can parse the screen and be like, what do I see? I think I'm in a room. What can I do? Are there any buttons I can press? I will press this button. That didn't do anything. Should I press another button?

It's wild. And you're like, what is happening here? And they can chart how far each model has ever gotten playing Pokemon, but it shows you this naive potential of what agents-- this is a very primitive environment. It's a very constrained environment. It's old school Pokemon. But it shows you the potential of what agents might be able to do.

So I would love to have an agent, to be able to have an AI agent and say, hey, book us a vacation next spring, and just like, have it know who we, us, vacation, spring, note what those things mean, and come back to either no clarifying questions or a couple, and like, hey, I got you, and your kids, and your partner plane tickets, hotel, activities booked. I actually replaced your luggage because I know that you've broken it down over the last 15 years, and you need a new roller bag.

So I got that on the way, and you're good to go and be like, wow. You can imagine that. We're not there yet. But I don't know how far off we are from that. And so that's going to be exciting as well.

STEVE MCGHEE: I hate to be bad cop here, Todd, but there's quite a lot of hope in your voice. And I recall hope being a phrase in SRE land, something we're not supposed to do too much.

TODD UNDERWOOD: It's weird. So people who think a lot about AI harms are some of the people who are most optimistic about AI benefits. And the analogy I'll give you is homeopathy.

No one worries that a homeopathic medicine will cause you harm, because it doesn't do anything. And since it doesn't do anything, it can't hurt you. It's just water, right?

STEVE MCGHEE: Just go for it.

TODD UNDERWOOD: Just go for it. I mean, you can overdose on water, but there's got to be many, many liters, right?

STEVE MCGHEE: Right.

TODD UNDERWOOD: With the AI, it's the same thing. If you think these things are incapable and aren't getting much better very quickly, then you don't need to worry about the harms because they're not any good. And so that's one camp of people.

There's another camp of people who are concerned about the human transition to these technologies, what the machines are capable of, what some of the harms might be during that transition and after. But those are the people who are seeing the improvement curves and saying like, it's hard in the ever present now for us to take this seriously, but cast your mind back 18 months ago. What could the models do? Cast your mind back 36 months ago. What could the models do? What do the models do today?

Even if you just look at the model launches that were in October of last year and the Gemini 2.5 and the Sonnet 3.7 launches, and you only think about coding. You're like, you know what writes really good code is Sonnet 3.7. And what writes really good code as of last week is the new Gemini launch. Those write really good code, and they didn't four months ago.

So I don't know what happens in 4 more months, and I don't know what happens in 12 more months, but there has been no plateau to the improvement. And therefore, I think, A, it's appropriate to be optimistic because you're looking at a curve, and you're seeing where it goes. But then B, it's also appropriate to be thoughtful and cautious, and say, OK, what are we going to do to make sure that this doesn't hurt people, this doesn't hurt societies, it doesn't hurt economies, it doesn't hurt the world? All right.

MATT SIEGLER: All right. I got a hard hitting one for you, Todd.

TODD UNDERWOOD: All right.

MATT SIEGLER: Some of our audience is listening, and some are also watching. Some of those who are watching only see the top of your shirt. Explain your shirt. Describe it out loud, and then describe the whole thing. What does it say?

TODD UNDERWOOD: Yeah. So the shirt has two goofy SRE logos, a dragon and a unicorn. The dragon is the original Google SRE logo. And the unicorn is the SRE EDU-- how do you say it?

MATT SIEGLER: Mascot.

TODD UNDERWOOD: Mascot. Thank you. I was like, mascot, that means pet, but we don't call it that anyway. I got stuck in the middle there. So the shirt says traitor and a scoundrel. And it's got a box filled in-- what kind of SRE are you a traitor and a scoundrel to? And I am a traitor to ML SRE as of 2023.

STEVE MCGHEE: So what had happened there? Why did we--

TODD UNDERWOOD: That's when I left. And I think, when people would leave my teams when I was a manager at Google, I would always call them a traitor and a scoundrel, because I love that people would go do stuff. I thought it was great. In any good technical environment, people go try stuff.

And then if they think your team is great, then they come back to something else, maybe, for a different reason. And that's healthy, and that's good. And so I was joking around with people. And then I worked with Skye Wilson. And we had the shirts made up, she designed it, and just actually started giving them to people, writing in like, I am a traitor to you.

I have one that says ML SRE because I founded Machine Learning SRE at Google. I have one that says Pittsburgh SRE because I used to be the Pitt SRE site lead, and ultimately, the Pitt site lead. So yeah, I really enjoy those. And I think it's a nice marker of where you've been.

STEVE MCGHEE: Yeah. I have another hard-hitting question, Todd. Now is your opportunity. You can do this if you want.

TODD UNDERWOOD: OK.

STEVE MCGHEE: Would you like to be like Niall and tear up your own book live on camera?

TODD UNDERWOOD: No.

STEVE MCGHEE: No. But if you were to, if you were to write another book, magically, it just happened, how would you change what you guys wrote about? What would be radically different? What are the diffs? What's the addendum that you would like to exist without actually doing the work of making it exist?

MATT SIEGLER: What do they call this, a fast follow or maybe just an errata?

TODD UNDERWOOD: Yeah.

STEVE MCGHEE: Yeah, that one.

TODD UNDERWOOD: Your experience with vibe coding, Steve, is where a lot of us are. And I think that getting the timing would be difficult. But what I would probably do is start-- no, but don't let me talk myself into that. If anybody's listening--

STEVE MCGHEE: This was not a trick. You're tricking yourself here. This is not--

TODD UNDERWOOD: Anyone is listening from O'Reilly, I am not writing another book. But if I were to, I would spend this next 6 to 12 months watching that transition and trying to plot out, what is our work like in a year to five years? Which is hard to do.

But I would say, what are the domains of expertise? How does testing and rollout work? Think about rollout. Most places don't have something like a canary analysis server or something that-- but most people have primitive smoke tests.

But what you really want is to ask a model like, I rolled out a thing. Does it work? Is it good? It'd be amazing to be able to say, I have some sense of what work and good mean. I think there's going to be a bunch of stuff like that. And so what I'd like to do is try to help understand what is going to be the technical work that we will have in the future, where instead of doing all of this, we direct the work of all of this.

In the same way, it used to be like you wrote code by hand with your fingers in an Emacs or VI window, like God intended. And now, you just say like, hey, could you unroll this loop?

Now, you know what unroll and what loop means, but you don't have to unroll the loop, it just unrolls the loop for you, or it makes the UI for you, and it plumbs them into your methods. So I think, that's what I would try to do is, I would try to say, what is it like to do technical work in a world where the execution becomes less and less important but the architecture, and the purpose, and the design are still important? Because I think that's where we'll be for a while.

Now, eventually, we will be past that, but I just don't know how quickly that will be. So that would be the book. It would be like, how do you direct the technical execution of a production engineering environment whose execution is managed by computers?

STEVE MCGHEE: Well, thank you, Todd. That was awesome. Thank you for your time. Is there anything else that you want to talk about that we didn't hit? Did you have a big gotcha that you wanted to just drop on the world?

TODD UNDERWOOD: You guys are covering great stuff. It's fascinating. One thing I will recommend people go look at the closing keynote of SREcon, which should be out on video by the time you guys launch this, because it's Charity Majors, Honeycomb, who has been notably AI-cranky, AI-skeptic, who is now AI-bargaining, so interesting transition there.

STEVE MCGHEE: Yeah.

MATT SIEGLER: We should include those in the show notes.

TODD UNDERWOOD: Sure.

STEVE MCGHEE: Yeah, will do.

TODD UNDERWOOD: Thank you.

STEVE MCGHEE: Thank you, Todd. It's been great. Nice to see you.

TODD UNDERWOOD: Good to see you.

STEVE MCGHEE: Until next time.

TODD UNDERWOOD: 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.