SDS PODCAST EPISODE 754: A CODE-SPECIALIZED LLM WILL REALIZE AGI, WITH JASON WARNER

Jon Krohn: 00:00 This is episode number 754 with Jason Warner, co-founder and CEO at poolside.

00:27 Welcome back to the Super Data Science Podcast. Today we are extremely fortunate to have the exceptionally gifted, and exceptionally visionary, Jason Warner on the show. Jason is co-founder and CEO of poolside, a hot venture capital-backed startup that will shortly be launching its code-specialized large language model and accompanying interface that is designed specifically for people who code, like software developers and data scientists. Previously, Jason was managing director at the renowned Bay-Area VC Redpoint Ventures. Before that, he held a series of senior software leadership roles at major tech companies, including being CTO of GitHub.

01:02 Today's episode should be fascinating to anyone keen to stay abreast of the state-of-the-art in AI today and what could happen in the coming years. In today's episode, Jason details why a code generation-specialized LLM like poolside will be far more valuable to humans who code than generalized LLMs like GPT-4 or Gemini. And he also fills us in on why he thinks AGI itself will be brought about by a code-specialized ML model like poolside's. All right, let's jump right into our conversation.

01:28 Jason, welcome to the Super Data Science Podcast. It's great to have you on the show. Where you calling in from today?

Jason Warner: 01:34 Thanks for having me. I'm calling in from my home in Victoria, British Columbia, Canada.

Jon Krohn: 01:39 Nice. And you're not Canadian originally, but you have found my home nation and fallen in love with it. I could add something I said to you before we started recording, is that if I ever moved back to Canada from New York, the West Coast, like where you are... So I always say

Vancouver, which is on the mainland, but you're on Victoria, which is slightly confusingly on Vancouver Island.

Jason Warner: 02:05 It's a great place to live, great place to raise kids. It's a wonderful place to be. And yeah, I'm an American, but though I'm now a Canadian permanent resident, and if I'm traveling abroad, I'll say, "I'm a Canadian." It's easier in that way, but yeah, people who don't know it, Victoria is a special place.

Jon Krohn: 02:22 I'm going to have to check it out someday soon. It does sound amazing. So we know each other through Kyle Daigle. He was in episode number 730. He is the COO of GitHub, and you used to be the CTO, so you guys overlap there. And he highly recommended you as a guest. When I asked him if there's anyone I should speak to, he said, "You got to talk to Jason. He's doing such exciting things." I looked at what you were doing and I was like, yeah, we got to get Jason on the show right away. So you're co-founder and CEO of poolside, which was funded by, or is funded by, Redpoint Ventures.

03:01 So for a couple of years you worked at Redpoint in between GitHub and now founding poolside. Coming up on a year ago that you founded it. So co-founder, CEO, and the thesis, from what I've read, and you're welcome to correct me, but from what I've read doing my research, the thesis with poolside is that there is tremendous value in large language models and in the flow tools for developers. So I have a lot of experience using GPT-4 for code generation as a part of ChatGPT. And so you had this experience in venture capital. Obviously there's a market there. Fill me in, fill my audience in on why a tool, a large language model specifically for code like you're working on at poolside, can be better than the general approach that I get. And I'm quite happy with using something like GPT-4 in the ChatGPT interface.

Jason Warner: 04:04 So let's start with a different way of trying to describe this. So imagine, if you will, that GPT-4, which is, as far as I'm concerned, the gold standard. It is by far the best in almost every way possible at what it does. But let's just call GPT-4, for the moment, the Toyota Camry. It is a vehicle. It is the bestselling sedan in the world, and is a general purpose vehicle. It can take you to work, go on vacation, haul your family around, go get groceries. But imagine all of a sudden, because it's the only vehicle in the world at the moment, you start abusing it for things that it really wasn't built for. You put a tow hitch on it, you start to pull loads, larger and larger loads over time. And for necessity, you start using it on the farm or on the job site, or you start tuning the s*** out of this thing and start racing it on the racetrack.

05:09 That works because it's the only thing out there. And then you have other people coming into the world building other versions of sedans saying they're slightly different. So Anthropics building the Honda Accord, open source models might be the Hyundai Sonata or whatever version of these things that exist. And developers are the ones who are putting the tow hitch on this, or trying to have it haul tons of hay, or using it on the job site, if you will. Well, we're introducing a new vehicle type. So still a large language model with applications that are built on top of it, but it's a new vehicle type, and it's the Ford F-150. And it is specifically built for those environments and those orientations and those jobs. And you could still abuse us. You could say, "Hey, well I'm going to take you on a 10,000-mile road trip."

06:03 I'm like, "Hey, you know what? That's not what we're for." Yeah, we're going to be capable of doing that, but you might be more comfortable going to get the newly introduced other vehicle type, like the minivan. And that's what's going to happen in the world over the next couple of years is that not one vehicle type is going to exist to serve all purposes. You're going to start to have specialty vehicles introduced. Now we happen to believe that we are not a fully specialized vehicle, we're more of a general special purpose, like a truck is a very different general special purpose than something like a dump truck, as an example. So that's how I'm trying to explain to the world what we're doing. These general purpose models are all sedans. We're introducing the world to a truck for the very first time, and that's what we're doing.

Jon Krohn: 06:51 That is an amazing analogy, and you can tell that you have a lot of experience either listening to you or giving pitches, because that was probably, of the hundreds of guests that I've interviewed on the show, that was probably the clearest analogy describing what somebody's doing in their AI company that I've heard. So, crystal clear. You're building the F-150. When will my listeners, roughly, be able to get their hands on an F-150?

Jason Warner: 07:21 Q2 of this year. That's when we want to launch.

Jon Krohn: 07:24 Oh wow.

Jason Warner: 07:27 This is a sprint now for us to get there. We're working with design partners right now, and we're showing them the applications that we're building on top of everything, and we're in the middle of model training at the moment. So Q2 is our target. Who knows? We just... When this is being recorded, things get announced. Meta just announced that they're trying to soak up every single GPU in the world at the moment. So things might change a little bit, but that's our target, Q2 this year.

Jon Krohn: 07:58 Very cool. Well that's exciting. And are you able to let us get a glimpse as to how modeling outputs might be different with an F-150 versus trying to use a Camry for code generation? Or are you able to give some insights for us into what the user experience would be like using poolside as opposed to using a ChatGPT interface?

Jason Warner: 08:22 A bit.

Jon Krohn: 08:23 Makes perfect sense. Totally understand.

Jason Warner: 08:26 And so what we're trying to show is that there is a moment in time that we exist in right now, which is why analogy like the Camry to the Ford F-150 makes sense, because we are all entering this land at the same time, at the same starting point, but we're diverging in the woods. The further we diverge in the woods as we walk these different paths, the more obvious what the terrain differences start to look like. But we are literally only a couple of steps into this park together. OpenAI has a couple of, let's say, a couple meters on everybody else at the moment, but the point being, we're looking roughly at the same landscape. From an experience perspective, what we care about is the future of software. So let me break down how I think about product development, and we'll get to why I'm going to answer the question this way.

09:23 I believe, similar to what Jeffrey Moore said a long time ago, there's areas of your product in the moment and then long-term that you think about differentiation, neutralization, and then optimization. Our model out of the gate will be differentiated. It'll be a different experience. You're going to feel it differently when people will use the model. In conjunction with the applications we build on top of it, which is Copilot or ChatGPT, we're going to neutralize more than not on that experience. And then optimization for us as a startup is not something we think about on a regular basis. That's for the big guys to think about. So our model will be differentiated. Our experience out of the gate will be more neutralized. So if you used GitHub Copilot, you kind of understand what we're building. If you use ChatGPT, you'll understand our web interface. Perplexity is probably closer to how we're thinking about our web interfaces, but you get an idea of what to experience there.

10:17 From a model perspective, there's a massive, massive difference between us and others in that, one, something that's tuned for software will know more about software. And what I mean by this specifically is, if Anthropic and OpenAI... OpenAI has made famous reinforcement learning via human feedback. Anthropic has made famous reinforcement learning via constitutional AI or algorithmic AI and things of that nature. We're introducing something that we call reinforcement learning via code execution feedback. So taking advantage of the aspects of software and the aspects of code that you might imagine. One, it's inspectable, two, it's runnable, and three, you have to compile or execute these things and you can get deterministic feedback. And so what we have done is inside of our training set, we've made very, very, very, very different decisions than general purpose models would make.

11:08 And this goes to the heart of why a truck is different than a sedan is, we've made very different design decisions. We have included only high-quality code in the model. We have cut data sources out of the initial dataset because it's a very different audience that's going to use this for a very different purpose. And so this goes to the reinforcement learning side of the fence, too. We care very deeply that yeah, we'll have human feedback as well, but the reinforcement learning is going to be all about what's produced from the software side. So we've taken about 50,000 real-world projects out, high quality real-world projects out of the initial training dataset and are using it on the reinforcement side. We've made all the git commits executable. We have all three legs of the stool that we need. We've got the issue that's described in real language. We've got the code, we've got tests, we've got all of the things that we need, and we send this through a reinforcement learning platform on our model to see what it does.

12:03 And we start to make our model better. We effectively are teaching our model how to code sequentially over time. And this is how you're going to experience the difference in here. It's just going to be orders of magnitude better on one. If you just think about this as code completion, code suggestion, it'll be better. But more than that, it's going to know more because it'll have experience more of not just code, but software, the entirety of software. That said, there's one even further unique difference between us and how most developers in the world will experience this at some point, because most developers in the world work inside corporations. And people who use GPT-4 today inside hobby projects or online, we're going to have a version of that as well. It'll be poolside, you'll be able to use it in a SaaS offering.

12:50 Our platform will be out there, people can move over. Repla might want to switch to our model at some point. All these online tools might want to switch to our model at some point, but we're also going to bring our entire stack, the model, the platform, the applications, to enterprises inside their environments to fine tune on their data. So all their code bases, all their knowledge repositories will be fine-tuned in a way that we never see the code. We never see these things. They don't make it back to poolside, they never make it back into our models. It's bespoke to them. It's all for them. And we do this, obviously, because that's where every organization in the world is going to get the lift. Our model will be the best in the world for software development, but contextually now it has all of your information. It's aware of your code names, your coding styles or your own homegrown programming languages or your customized web frameworks or any of those sorts of things.

13:49 Heck, at some point you're going to change your policy management to hook into poolside because that's going to be how these things work in the future. So I just spoke for a while trying to explain how people are going to feel differently when it manifests in the world, but it goes to why this should exist in the first place. The model will be better because it's specifically oriented for software development. The go-to market will matter because now finally, for the first time, enterprises can actually use one of these things in their environments. And make no mistake, enterprises should not be using models that are ever going to train on their outputs. And thirdly, developers are going to finally get access to these things in enterprises.

Jon Krohn: 14:27 You just did my job there because my job is to summarize back to you what you just said, but you just did it perfectly. So tick, tick, tick. And so then I'll just say that I love... I can see how from an investor's perspective, having that enterprise play at the end there, fine-tuning on-prem on their data, that's going to be nice and sticky. And it's a brilliant, brilliant strategic choice to make there in developing your product. And I'm sure your investors will appreciate it as well. Awesome. So loved that introduction to poolside. It occurred to me as you were speaking that maybe I understand now the poolside name as well, which is maybe this idea of a developer sitting poolside, maybe with a cocktail or something, while code is being written for them and they can be enjoying life.

Jason Warner: 15:28 So I have a firm view... Let back up a little bit for the folks that aren't familiar with me as well, and I'll just give a little bit of background on me and a lot of this will make sense too, but you're dead on on the name, for what it's worth. My background is poolside. Before this, I was investor at RedPoint, as we mentioned in the intro, where I was investing in AI and infrastructure things and then incubated poolside to roll out. Before that, I was a CTO at

GitHub; two years pre-Microsoft acquisition, two years post. Before that, same thing at Heroku, which is a platform as a service now with Salesforce, one of the most popular [inaudible 00:16:04] in the world, even to this day. And before that was Canonical, the people make Ubuntu Linux. So my entire user base, customer base for the past whatever, 15 years of my life, has exclusively been developers.

16:17 And all I really think about is the future of software. And I think about theoretical possible workflows. Every time a developer touches a key on their keyboard to what happens in machinate and production. The theoretical best possible flow there is every time a developer touches a key, something happens in production, collapsing the entire chain down to as narrow, as collapsed a view as possible. And it's not possible to get to that level, but you can understand, when you think about the future of software, what it looks like. With poolside, with AI, with poolside in general, and what the world is going to look like in the coming years. And make no mistake, bad metaphors aside, this is literally the top of the first inning, first pitch pitched type of deal where we are right now.

17:07 And I don't mean we as in poolside, so I mean we as an industry, as a world with what's about to happen here. And from an investor standpoint, I don't think any investors really fully recognize this, because what I'm about to say might blow some people's minds, make them think I'm absolutely crazy. But we believe that we're in the world right now where we're effectively giving all the developers around the world AI assistance. So your developer-led AI assisted. And these AI systems are going to get better and better and better. They're going to have more and more agency. We may or may not give them autonomy, but we're basically going to sit there and say, "Hey, the world's best pair programmer or assistant next to you." At the moment, if you give a senior developer any of the tools on the market, it's like a power tool.

17:51 If you give a junior developer any of these, it actually could be disastrous. So there's this wide gap in between who is using these tools and how effective they are. In the coming years, what will happen is, as we add more agency to these things, and these tools themselves have to earn the trust of the humans involved, will be AI-led, human-assisted. AIs will start to do more and more and more. You could view a world in which, as an example, like what poolside might do in the future when I think about the future of software is, you log in to poolside inside your enterprise and all your Jira tickets, all of your GitHub PRs, all of your Valgrind runs or your debug runs, all your pre-production check flights, whatever internal systems and checkpoints that you use are all amalgamated into one view, and poolsides gone through and said, "Hey, I've taken some liberties on your side. I've decomposed a Jira ticket, or I saw somebody said they rejected your PR, so I've made a change. Would you like to resubmit it?"

18:55 So that's like AI-led, human-assisted. When you start to decide, the AI did something for me, do I want to go forward with it? And you can start to see how that's going to evolve, how that might feel to somebody. And this isn't about automatically writing documentation or automatically writing unit tests. All those things are features. Those things aren't actually real platforms. They're not real products. Those things will go away in a couple of years. Those are all feature sets of these autonomous AIs, and that will give way to, eventually when these AIs are so smart, that 90% of people in the world can write 90% of software. Not everything, but 90% of software can be written exclusively by someone just interacting in an NLP style with these AIs. We call that AI-led, human-assisted.

Jon Krohn: 19:37 I totally follow you. I buy into your vision. I don't think you're crazy at all. I think that that is absolutely where we're going with this. And then you've also now started to answer what was going to be my next topic area. You're segueing into it perfectly. This is a guest-led, host-assisted episode, because the next thing that I wanted to ask you about is, I read, and so this is a quote from poolside. It says, "poolside aims to unlock humanity's potential by pursuing AGI, artificial general intelligence, for software creation with the fundamental belief that the transitional path for humanity to AGI is by building for specific capabilities instead of a general purpose approach."

20:26 So you've now started to give us some of the picture, I think, into why you believe that, and that belief is, it's maybe not the most popular perspective from people who are developing AGI. So DeepMind, for sure, at least at this time, they are taking the tack of let's build a system that can do one task well, and then have that same algorithm do well on parallel tasks, and then let's see how we can get that to do on even less related tasks. So they are going from specific to general deliberately, and it sounds like you are taking the opposite approach. And so fill us in more on your thinking there.

Jason Warner: 21:12 So there's a couple of general thoughts here which lead us to this. And I'm going to be super transparent because the listeners here... I doubt there's too many investors that might be listening to this, but I think it's fun for people to understand how my mind works and how... Eiso Kant is my co-founder. And for those that don't know him, he should come on here and talk about some of the specifics that we're dealing with as well.

Jon Krohn: 21:36 I'd love that, for sure. And we also do... At least I know personally, at least in the New York area, I don't know about Bay Area or other parts of the world, but in New

York I frequently run into investors who listen to my show.

Jason Warner: 21:49 So investors, if I insult you in the next couple of minutes, I apologize. How's that? But here's our view of this, which is that... First, let's just conceptually say does the world need another sedan? Yeah, there's always room for more sedans. There's always room for another one of those. But do we have a unique viewpoint? Is it worth our time, energy, and effort to go do this? Our unique viewpoint was a new vehicle is needed. A new vehicle is needed because, one, of a market opportunity. Yes, 100%. We think of it as a market opportunity. In fact, we think that the software side of the fence is actually an unlimited TAM, just like the general intelligence side of this is effectively unlimited TAM. That allows us some liberties. One of the liberties is that we could self-fund this company to pursue AGI.

22:42 We don't have to rely on the magnanimity of these massive institutions to throw billions of dollars into us in that same way. We will have to build a real business, we will build a real business and we can self-fund this to a degree. That's the practical side of that answer. That's the very methodical, "we're going to build one of those businesses" side of the answer. The other side of the answer is we think it's actually at least as viable, and it's important that I state this, that we think it's at least as viable to start here. Now we also happen to go a bit further. We think it's a more viable, probably a faster cheaper, path. And the reason why is if you deconstruct what it means to develop software, think about what it takes to actually build software. You have to have elements of planning. You have to solve for elements of understanding and reasoning, and you have to be able to hold these concepts in your head. It's not literally just about all of the quote/unquote "next token" type of stuff.

23:40 You actually have to hold these things. Your context windows have to get larger because you got to hold it available to you in your head. Let's take AI out of it. Let's take LLMs out of it for a second and talk about the best developers we've ever met in our lives, and what do we always say about them? They can hold these things in their head while they're rotating them or while they're deconstructing them or while they're doing this, they can zoom in and they can change a couple of things and they can go back out and it could largely work well. What are we actually describing when we talk about those folks? And a lot of the papers that have been released recently, which is the geometric one from DeepMind as an example, they all point to the fact that when you have LLMs married with effectively some structured elements to use, you get a much quicker, broader point towards intelligence.

24:32 And so what is the area of the world that we have the most data on that has some structure to it? Well, my view it's software, it's code. You can talk about mathematics or physics, which are... And if you think about these from a pure, pure, pure, pure definitional perspective, math is all the way over here. Pure conceptual math is all the way to the right. Then you've got physics and you've got computer science and they're all sub elements of math. But we just happen to have all of this data on computer science, the most copious amounts of data in the world. And so that's part of our thinking there. It's exactly that. It's like we're growing into that area, it's conceptually true. Now we also hold some other views, which is, it's likely not going to come from this wave. We don't know the answer to this question, but it's not likely that you're going to get to that level in this wave.

25:21 We actually hold probably a little bit closer to Jan's thinking here, which is LLMs as policy moving on to a different architecture at some point. But the marriage of these things is actually what's going to get us to something reasoning these AGI elements. But that's one of those over beers at a conference start discussing these things and debating them type of view. But you can largely understand what I'm meandering around in this, the thought maze. It's by the very nature of what software is, it allows us to structure methodically get to these points.

Jon Krohn: 25:54 It makes a ton of sense to me. You've absolutely sold me on this. A quick abbreviation explanation there for our non-investor listeners. The TAM word that Jason mentioned a couple of times there is the total addressable market. So how much market is there for the product that you're building? And you quickly mentioned there, at the time of us recording this episode, the DeepMind alpha geometry paper had just come out. I'm planning on doing an episode dedicated to that, coming up soon, but I'll have a link to that fascinating research. So it's a huge advancement in AI reasoning and math. I'll have that DeepMind paper on alpha geometry in the show notes, but to now agree with what you were just saying, I totally get it. I hadn't thought about what you're saying before, but it makes perfect sense to me with what you're doing at poolside. So not only are approaches like alpha geometry from DeepMind, but also the rumored Q* algorithm from OpenAI.

Jason Warner: 26:52 So Q*, what we know about Q*, we have a lot of folks over at OpenAI, so we can fortunately verify this. Q* and our approach is actually very similar in a lot of ways. And you can think about deconstructing what Q* has been rumored to be about. You can understand when we start talking about planning and we start talking about AlphaGO style, like deconstructions of semantic trees, which are basically lines of code inside there, what we're going to experiment with. This is our version of research at poolside, autonomous code writing via these folded structures. This is what we're doing.

Jon Krohn: 27:32 That is absolutely fascinating. I didn't know that doing the research for this episode and wow. This has turned out to be an extremely fascinating episode. So if listeners want to hear more about Q*, I talked about that. I dedicated episode number 740 to Q* and what's rumored about it. And so we know, based on papers that OpenAI have been publishing that seem to be related, and that Jason is nodding his head to, and so it sounds similar to what y'all are up to at poolside, where you are deconstructing a multi-step problem into these steps, and figuring out what the best path to take is at each of those individual steps.

28:14 And so in some of the papers that OpenAI has published, which may or may not be related to Q* but seem to probably be, they had to create a dataset of hundreds of thousands of step-by-step answers to math questions. There was tens of thousands of math questions with hundreds of thousands of step-by-step steps, and those had to be created by a human. The advantage that you're describing with taking a similar approach to code is that you don't need to have... Because you already have it. You have way more abundant data, orders and orders of magnitude more abundant data. And because it's executable, you also know whether it's going to work.

Jason Warner: 29:00 Yip. When we dive into this and start talking about it, people do... The lights go on. And for us, again, it's one of those moments where we do sound crazy because we're taking a very different approach. But the entire purpose of what we're doing is we do think that we're zigging when others are zagging because, again, we're not, intentionally, we're not trying to build a sedan, we don't actually care about building the sedan. We care very deeply about building the truck. We think the truck is also a vehicle type that's going to serve a lot of purposes and eventually they might converge. The Ford F-150 has a Raptor version which you don't use on the farm. I'm abusing the analogy here, but you get what I mean by this. Software allows us a lot of things that, in a world... It allows us a lot of liberties and a lot of advantages. And I think that we have underestimated how much advantage we can get out of this, which is also why small code-only models don't make sense.

30:00 Small code-only models for code completion don't make any sense anymore because the point is for these things, unless you exclusively care about code completion, if you only care about code completion as a tool, code-only models make sense. But if you care very deeply about what the future of software looks like and the future of all of this stuff, you start to understand. poolside looks like OpenAI, just oriented towards a very specific domain, a very different domain.

Jon Krohn: 30:24 And a domain that happens to potentially be a path to AGI. Very cool. So I want to be conscious of your time, and so this is absolutely fascinating. We could frankly talk about it for hours, and maybe we will have Eiso on in the near future to dig into this a bit more. But before I let you go, there's another fascinating thing about what you're doing today that I'd like to just talk about with the audience a little bit, and this is that you sit on the operating board of one of the oldest and most renowned hedge funds in the world, Bridgewater. So what is an operating board? And what do you do on that? How does that help this huge renowned hedge fund be more effective?

Jason Warner: 31:15 So this is new. I just joined officially late last year, late in 2023. Bridgewater is the world's largest, oldest, maybe one of the most storied hedge funds of all time founded by Ray Dalio. And what the operating board is, is a way... It's not a public entity. Bridgewater is not a public entity. So what it is, is a way to mimic some of the public controls in a private entity as the company was transitioning from a founder-led organization with Ray, to an entity that wouldn't involve Ray in day-to-day business. So if you go and look at who is on the operating board, it's either the current CEOs or presidents of major institutions or former CEOs of some of the largest things in the world. Because it's all about operations, it's all about understanding how to put the controls in place and the right protocols, etc, of that.

32:08 My involvement with them is obviously going to be from a technical side of the fence and an artificial intelligence side of the fence. And Bridgewater is one of the probably most talked about in terms of the way that they've used computation to do trading for a long time, but this wave is different. They have an initiative that is all about large language models, about AI, about using and applying those initiatives to their domain. And then obviously it's just the pure technical side of the fence, too, is being a 20-year CTO scaling and operating technical business is what I've done, and then applying intelligence to them is what every company in the world is going to have to do in the next 10 years. And Bridgewater has a massive head start on that from most of the institutions in the world, but I lend a hand where I can on that.

Jon Krohn: 32:58 Fascinating. Very cool to think about and yeah, it can make a huge impact there for sure. Thank you, Jason. This has been... I knew this was going to be a good episode, but wow.

Jason Warner: 33:09 Thanks for having me.

Jon Krohn: 33:09 It's just incredible. Before I let you go, I ask all of my guests for a book recommendation. Do you happen to have one for us?

Jason Warner: 33:16 Ah, book recommendation. So I like to recommend some of the same ones a lot of people in Silicon Valley recommend, but there's one I like from Bill Walsh called The Score Takes Care of Itself. And I tend to think about this for people who are starting companies and operating them, which is you've got to focus on what matters and break it down. Everyone wants to talk about the score, but you got to talk about what leads to those positive outcomes. So that's a good one, but that's not the one I want to recommend. The actual one I want to recommend is one I've been recommending for years, because I'm an organizational dynamic person. What I've done mostly in my career is product development, but I've come into organizations and make them better over time. And there's this book called, I believe it's called Simple Sabotage.

34:03 And what it is, is there's a 1944 field manual from the, I believe, the CIA, which was used for people behind enemy lines on how they could subtly sabotage their enemies. So it was all about like, "Hey, if you're working for us while over there in that way, what do you do? Well, here's what you do in meetings. Here's what you do for process. Here's how you introduce bureaucracy. Here's how you slow things down," that sort of thing. And I think that we do this all the time in our businesses without knowing it. We're actually literally doing these things all the time. And I think it's worth everyone spending time reading this book, and to realize that this was a weapon to use against your enemies. So guard against it coming into your organizations as much as you possibly can.

Jon Krohn: 34:53 Wow, great tip. Great recommendation there. Simple Sabotage as well as earlier, The Score Takes Care of Itself. Jason, this has been a mind-blowing conversation for me. I'm sure a lot of our audience would love to know where they can follow your thoughts after this episode. What's the best way to follow you?

Jason Warner: 35:11 Twitter. Jason C. Warner on Twitter is probably the best, although you're going to get a mix of snarky comments, family updates, and then 30 tweet threads on how to make decisions inside companies.

Jon Krohn: 35:24 Nice. It sounds great. Jason, thank you so much for taking some of your precious time to speak to me and enlighten our audience. It's so exciting to see what you're doing at poolside, and hopefully in Q2 we'll be able to check it out ourselves or not long thereafter. It's really tremendous what you're up to, and I can't wait to hear how the journey is coming along later on.

Jason Warner: 35:46 Thanks for having me, Jon. Appreciate it.

Jon Krohn: 35:48 Jason's ability to convey massive transformative ideas so clearly and concisely was inspiring to me and something for me to aspire toward. I hope you found today's episode as fascinating and exciting as I did. Jason covered how the heavy duty Ford F-150 LLM that poolside is training will be markedly more valuable to humans who code than the general Toyota Camry sedan of GPT-4, Gemini, and other generalized models. He also talked about how we'll shift from a developer-led AI assisted paradigm to an AI-led developer assisted one in the coming years, and how the wealth of executable multi-step logic code available to train an LLM on could be the shortest path to realizing virtuoso AGI. Wow, what a mind-expanding concept.

36:30 That's it for today's excellent episode. If you enjoyed it, consider supporting the show by sharing the episode with people who you think might like it, reviewing it on your favorite podcasting platform or YouTube, or subscribing if you don't subscribe already. But most importantly, we hope you'll just keep on listening. Until next time, keep on rocking it out there, and I'm looking forward to enjoying another round of the Super Data Science podcast with you very soon.