

1069 - The New, New Kingmakers: Agents, Developers, and the Future of Software with Stephen O'Grady

[00:00:00]

Dark Web Exposure

Aaron: What does the outside world already know about your company? Cybercriminals could be seeing leaked credentials, compromised session cookies, exposed infrastructure, and even impersonations of your brand and executives. NordLayer Intelligence by NordStellar gives security teams visibility into those threats across the deep and dark web, data breaches, attack surfaces, and brand impersonation all in one platform.

Find out what attackers know before they can use it. Visit nordlayer.com/intelligence/enterpriseai and use code enterprise ten for ten percent off your NordStellar plan

Identity Security Pitch

Brian: Today's show is sponsored by Palo Alto Networks. Every time your team deploys a new cloud workload [00:01:00] or AI agent, another identity gets permanent access to your critical systems. Legacy tools were built to manage human employees, leaving modern machine and AI access largely unmanaged. That's why Idira by Palo Alto Networks comes in.

Human, machine, AI, one identity platform for all of them. Idira replaces permanent permissions with dynamic access, so you can lock down every identity without slowing down your business. Secure every identity with Idira by Palo Alto Networks. Visit paloaltonetworks.com/idira. Again, that's paloaltonetworks.com/IDIRA.

Meet Stephen OGrady

Brandon: And we're back. Joining me today is Stephen O'Grady, principal analyst and co-founder of RedMonk. Welcome to the show, Stephen

Stephen: My pleasure

Brandon: Well, listen, I don't think you need a long introduction. I think people already know who you are, and we're gonna talk a little bit about your your great book, The New Kingmakers, and your recent response to your own book, if you want a second.

Walking Fitness Routine

Brandon: But before, [00:02:00] before we jump in, I thought I'd start with something kind of fun. I... Last I looked I've seen on the social media you're, like, a big fitness walker. You've been, like, walking many miles and, you have a bunch of blog posts about it. So I was gonna say are you still, I know you've got family now, priorities and stuff, everything. Are you still getting your walks in? You still making it around your little area out there up in Maine?

Stephen: I am. I am still still at it. It's, I got into it a couple years ago when I was like tried to get back into running and just kept like, "Oh, my foot hurts or back," or, everything's tweaked, and it was like, eh, just walk a lot, I am fortunate to be in a profession where I can do a bunch of things.

Certainly we'll get into the a bit later, but even taking, listening to calls and all that, I can do while I'm walking around. So it's a good way to combine in many cases getting some work done with getting some exercise

Brandon: Yeah, no, I think, I agree. I think it's a great way just to... I always, I tell people don't sit at your desk and watch the keynote. Put it in your phone and then walk around and watch the keynote. You'll stay more involved. I did say, one thing that I thought, did you walk something?

In, I remember seeing a picture of it was, like, very cold, or it looked very cold. It was, like, snowing, and you did like a long eight-hour walk or something. Is this, am I [00:03:00] remembering this correctly, or?

Stephen: yeah, I've done walks that are cold. I've done walks that are pretty long. I did 50 miles one time, 60 miles another time.

Brandon: Like a serious walker. This is-- I don't want people to be like, "It's just I'm just out like taking a stroll." Like you're doing the real thing,

Stephen: No, but honestly, I tell people all the time, man if all you can do is get out and take a stroll, like great, man. Like any, any movement's good. So if you got 20 minutes, half hour, whatever it is, recommend it

Brandon: Do what you can do. Hopefully we always get notes that people are running, walking, wa- walking their dogs about the podcast. So hopefully people are out right now. They're listening to us while they're on a, a great new walk.

Why New Kingmakers

Brandon: Listen as I mentioned at the top there, so you wrote a book in 2013, *The New Kingmakers*, which I think was like, I don't know, I think it's like a seminal book for the industry, took a lot of ideas at the time and kinda condensed them into a nice narrative for all of us.

And but of course, many things have changed there. So I thought maybe let's just start with for those that haven't read the book, maybe why don't you set it up? What was-- Take us back to 2013. What were you thinking about? What prompted you to write the book, and what, kinda what's the core thesis of it?

Stephen: So what prompted [00:04:00] me to write the book was that I got tired of having the same conversation over and over, and I thought, hey, if I document this in a short book, 'cause I don't remember how many pages it is, but it's something like eighteen, twenty thousand words. So it's not a sort of lengthy tome.

But basically the, the gist of the argument was that, for decades in the technology industry largely as a function of budget and purchasing and all that developers were essentially an afterthought, right? It was like, okay I'm a CIO, I'm gonna go buy some things and this is what developers are gonna use.

And, over a period of years a bunch of different things came together to change that. Open source being one cloud being another. There's a whole range of other things we could talk about, software as a service and education sort of marketplaces for services and so on that basically empower developers.

And, by twenty thirteen when the book came out the reality that we were seeing was materially different, which was basically that excuse me, in many cases, when you looked around in the industry the technologies that were actually being used were the ones that developers wanted to be used.

And we basically just came up with a I wrote the book to [00:05:00] essentially crystallize and encapsulate that argument, like I said, hopefully shortcut the conversations I needed to have.

Brandon: It makes sense. And so what do you think? What was it about this time period? Why did developers go from being, if you will, being directed to and said "This is what you're gonna use," to like this is, for them basically taking an active role in choosing technology. What do you think was like...

What- is there one reason or is it a whole s- host of reasons?

Open Source Empowers Devs

Stephen: It's a host of reasons, but probably the, the sort of biggest one was open source, right? So in other words, when I was coming up in the late nineties if you wanted to do anything as a developer, right? If I need an operating system, I needed a database, if I need... basically, I wanna go out and build something the early sort of versions of MySQL around in the late

nineties Linux was coming up, but, by and large you would have to go and get budget from somebody, right?

You would have to go and get budget from somebody to, to purchase license for an OS, purchase license for database and all that. And, what ended up happening was that with open source, they didn't have to do that anymore, right? And, all of a sudden it was, "All right, I wanna go build something," and [00:06:00] there's no sort of commercial restriction me doing that.

I can go pick up a quality free operating system, a quality free database. Languages obviously were free to acquire and even tooling. Tool, developer tooling for years had cost a ton of money and, So anyhow you had this situation throughout the early two thousands where developers were empowered in ways they just fundamentally hadn't been before.

And then in '06 what is it? In March and August of that year S3 and EC2 launch kicking off the cloud wave. And all of a sudden, not only do I have free tools or free software the free as in beer software, I, all of a sudden I have hardware that is affordable, right?

I can go run, a cluster for twenty minutes or an hour or a day and, we're talking in sort of dollars oh, that might cost me twenty bucks, right? I don't need to come up with thousands of dollars for to go out and get sort of physical hardware. So anyhow, I think, like I said, I think the biggest thing for me was open source.

And I remember as an analyst when I first started talking to another analyst from a commercial who was responsible for surveying commercial database markets, and [00:07:00] I had asked "Hey, what about MySQL? What about Postgres? What about these sort of open source databases?"

And they said, "Oh those aren't paid for, so we don't track those." And that just, very early on, like I said, my first days as an analyst, I was like, "That doesn't make any sense, because that's not gonna reflect usage." All you're measuring is what people are paying for, and if people are using these things that aren't being paid for, then, we need to track that.

So anyhow, those are the origins, and then that progressed over sort of a decade plus, and like I said, there's many other factors that came in, but,

Brandon: Yeah, that makes sense.

Stephen: that's what started it.

Linux and LAMP Takeover

Brandon: But I guess when I think about the, this, this book and I think about the software, I think, I guess I just think Linux was like the driving factor, right? I think that was the thing that I don't know, maybe crystallized all of this, right? Was the developers essentially saying, "Not only do I want to use this, I'm going to use this," right?

And almost dragging Linux into the enterprise to the point that, then we had Google, of course, prove that commodity service, servers running Linux was a very powerful way to run large websites and Red Hat. So I don't know. I guess when I think of it, I almost feel like without Linux, like this maybe would [00:08:00] not have happened.

It's like that was the linchpin that really broke it open. I don't know. Do you think of it that way?

Stephen: Yeah, I think Linux is certainly key, but honestly I would expand that a little bit. And honestly, I think the biggest, example that people point to is the LAMP stack, right? So it's an operating system, a database a runtime, and a web server, right? And that just, that sort of package combination just ended up everywhere.

So it's, no matter what you're building, no matter how you wanna build it, it's gonna give you some of the requisite pieces to go out and build whether it's a, a toy app for yourself as a developer, whether it's, I remember talking to a bank and, they first engaged with Red Hat this was years and years ago, obviously.

Because they walked into their data center and they're like, "What is all of this?" "I didn't know we were running this." And Marten Mickos, back when he was the CEO of MySQL, loved to tell the story of, walking into, again, I think it was a- another bank on Wall Street and saying "Hey, you folks might wanna talk to us about some MySQL licensing."

And they said, "Oh, that's open source. We're not using that." And, he had the wherewithal to say that's interesting 'cause it's been downloaded from, your IPs [00:09:00] five thousand times. So it's probably in there somewhere." So anyhow, yeah. I think, Linux is, really at the core of all of this for sure.

But I think there were other pieces that need to come along with it.

Brandon: Now for sure the LAMP stack.

Why Sun Failed

Brandon: And I think but I also think of the company that maybe was, like, largely displaced, at least in the most meaningful way, was Sun, right? Sun went away. And may-maybe we'll do a quick aside here 'cause this is a fun topic that we all like to talk if you've, of a certain age.

Brian Cantrell recently revisited his why did Sun fail, right? And I think, there's only maybe 40,000, 50,000 opinions about why Sun didn't make it through. But I don't know, what's your opinion? Why, what, my opinion, to put it on the table, is like I think there's nothing.

I don't think there's any, like, magic thing you could've done. I think when you if you will, it's like a tidal wave. When the tidal wave shows up and takes away all your revenue at least for your big revenue you're just in, you're just, it's, there's really no solution. So I'll put my cards on the table.

What do you think? Was there some magic thing Sun could've done? Wave your wand, go back in time. What would you have advised? What, or maybe you did. Maybe you gave the advice at the time. Who knows?

Stephen: No, I think like every company... first of all, I think we should back up and say the, the history of the industry [00:10:00] is that there are very few companies that have survived waves of disruption, period. That's just not a... Most companies are not able to do that.

And they're not able to do that for a variety of reasons. Clayton Christensen wrote a-- thought about this in "The Innovator's Dilemma" and so on. Sun I think was, a, a difficult case in that it was a bunch of sort of disparate businesses, right? In that they had the ultimate scale-up business, right?

From a hardware standpoint. They obviously had the sort of operating, you know, core sort of operating system. And then, they had the the runtime business, in Java. And, trying to have all of those weather...

So in other words, on the hardware front you have the disruption from scale-out, right? So Google and everyone else is "Okay, we're not buying bigger machines. We're buying tons and tons of cheap, smaller machines." The operating system and the, the runtime businesses are facing different disruptions, right?

Obviously Linux is coming in and disrupting Solaris. A host of languages were flooding in to disrupt Java. Ultimately, I think it's, I don't think there's any sort of one cause. I think it is, it was largely a [00:11:00] business that was phenomenally successful in one market and then just faced, kind of disruption from every potential quarter, right?

And that's, that, that's just, it's very difficult to navigate that.

Brandon: Yeah, I think it's a fair assessment.

Oracle Cisco Stickiness

Brandon: Now Sun sort of being the big loser, if you will, at least one of them that lost out. Now the, the... If I had gone back in time and you'd asked me, 'cause I think at the time Sun bought MySQL, and I really, at that point, I was like this is gonna be the end of Oracle," right?

Oracle is gonna go the way of Sun. Obviously, that hasn't happened. And then I think of also Cisco is g- related, not related, right? That was a- another area where I was like I don't know," open source networking and this is getting simpler and things like that. Are you, looking back on it, are you surprised those companies have generally, doing well?

I guess you could put that depending on quotations and exactly what you, how you wanna do it. But they weathered the storm. Looking back, would you have made that prediction at the time, or are you surprised that they weren't as disrupted at least as significantly as like a Sun?

Stephen: It's a good question. The Cisco thing doesn't surprise me as much. Networking was very and arguably still is very is [00:12:00] largely opaque to and inaccessible to developers. At least it was up until Open Compute and so on. But largely because it's not something that developers spend a ton of time and attention on themselves-- they just count on a network stack, and they take it for granted, and there's not a lot for them to geek out on, right?

In the way that there are in, operating system, database, et cetera, where these are tools that they're gonna leverage themselves. So the Cisco side of things doesn't surprise me as much. Oracle, I think you, you could make an argument that, yeah, that's a little more surprising, but the...

I always think back to talking to reporters over the years as an analyst. Every, every time there was an Oracle price hike, some new, reporter would show up at our doorstep and "Oh, my God, did you see Oracle hike their prices by X, Y, or Z?" And, "Is everybody gonna leave now?"

And we'd have to say, "Whoa, whoa slow, slow down." It's one thing to migrate, other pieces of the stack. The database is the highest risk asset that you have. And, even in cases where I can't remember how many years it was, but it took [00:13:00] Amazon years to get off their Oracle databases in spite of running obviously their own very competitive and stable and successful database as a service offering, right?

So yeah, I think databases tends to be stickier, I think, than people imagine. And so yeah, would I have been mildly surprised? Possibly. But again, it was, I don't know. E-even at the time, people, I remember all these forecasts like, okay, open source databases are gonna go out and displace all of these, these commercial businesses.

And I can't remember what year it was, but might have been '08 or '09. Anyhow, I was at the open source business conference that used to be run by Matt Asay. And they had a panel of, banks and insurance companies and media companies, all these like Fortune 500 CIOs. And they were talking at the time about, hey, what are you doing with open source?

And they were using open source everywhere, right? It was like languages, tools, runtime, operating system. And, with one or two exceptions, they're all like, "Yeah, we're not on the open source databases yet. We're just, we're not." So yeah, I guess the Oracle side of things [00:14:00] doesn't surprise me as much.

Brandon: Yeah. Some things are just stickier than you think. I think VMware is the latest thing. People are like, "Oh, VMware is gonna be displaced." And it's like, no. It seems like it's doing pretty well still so as we go forward here you dispensed this advice for probably, I don't know, at least a decade before there were big changes.

Did Execs Listen

Brandon: Just what's your take? Did executives d- largely listen? I'm sure there's cases of, people listened and people didn't. But do you think overall, like, when you're talking to the

companies people took it to heart and embraced developer... And obviously developer relations is goes through ebbs and flows, but that's certainly something that has been more, maybe more prevalent than it was before.

What's your take? Did people listen to you?

Stephen: Yeah. I don't know they listened to me, I think so much as they-- it just became unavoidable, right? In that you, I can't remember. It was a Red Hat event, I think. I don't know what year it would've been, like twenty fifteen maybe, sixteen. The CIO of Barclays Kieran Broadfoot, I think it was, was at Barclays was saying like, "Hey, we hand this book to everybody coming in," right?

You have conservative, very large organizations like SAP, talking about the new key makers at the time. Certainly there was [00:15:00] some like, hey, I, I get, having a name and having a vocabulary made it easier to talk about, I think.

But, really, I think what ended up happening was is that the book just reflected a reality, which was, all right, developers are just much more important than they used to be. They're much more valuable. And they are not, for years it was bit-- they were treated kinda like factory workers with pocket protectors and, it's a fungible resource.

And all of a sudden it was like, "Oh, wait, no, this is a competitive differentiation." And, looking around the industry, it's, like think of acqui-hiring, right? It's like, okay, I'm gonna go out and pay a premium for the startup. I don't even want the tech, I'm just hiring the team. And, that, I think that was the, maybe the most extreme example, of the value of this persona, if you will.

So yeah, I don't-- like I said, I don't know that, we or I or the book persuaded sort of anybody at scale. I think people, kinda looked around at the reality around them and said, "Okay, yeah, no, this is, yeah, this is a very important audience that we need to court."

Brandon: Yeah. I've always thought this book pairs, I don't [00:16:00] know, maybe intentionally, unintentionally a little bit with Marc Andreessen's "Software Eats the World" or "Eating the World," whatever that was. And it's like you can almost just give somebody like these two things. There, there's-- It's a slightly different take, but it's in the same area, right?

It's And if you didn't, I feel like if, as, as an executive, if you didn't take some of that to heart, I don't know, you probably didn't do well. May- maybe you did. May- maybe

Stephen: pretty sure-- It might have been so long since I read it. I'm pretty sure that is in fact referenced in the book, right? And that was again just the idea that, for those who are unfamiliar with it, it's like, hey, software businesses are making intrusions into all these traditional categories, right?

He goes through the examples of the-- Brian Cantrell loves to point out that if you go back and look at the actual examples from the essay the examples themselves haven't aged so well. But the, the concept itself is, I think and, the idea is an accurate one, right?

Which is that digital, businesses were really becoming challengers, and these digital businesses were run and fueled by developers, right? And yeah. And I think it's just another example of how important that audience is.

Brandon: Absolutely.

LLMs Arrive Suddenly

Brandon: All right, so we're all cruising along here. We're all, [00:17:00] you're out doing stuff. We're all building software. Then of course, the world starts to change. So maybe the, the simplest start is on the analyst side, were you already tracking LLMs? I always think of LLMs and I guess the, the date is like the release of ChatGPT, right?

When it went I guess whatever it was, 3.5 or whatever it was. Like that's the demarcation line I like to use. Were you immersed in this world at all? Did you see this coming in a way that maybe others, at least my- myself, didn't necessarily see it coming?

Stephen: No. So I think there, so there's two ways to look at that. The short version is that do we see the world that we're living in right now? No. Right? And I think, there are so many times over the years at Redmonk where it's like, I, I'm just going through technologies like open source or SaaS or even something like distributed version control.

It's like, okay, th-this is a thing. This is where the puck has headed. It's not hard to see if you're talking to the right people. But then, LLMs just burst on the scene. We heard, probably like yourself and probably a lot of people in the industry, we'd heard the behind-the-scenes rumors of "Hey, man, there's some tech at these companies [00:18:00] where it's borderline creepy," right?

Where it's like, hey, you can't really tell if it's human. I think can't remember if it was somebody from Google. A-anyhow, there are all these sort of rumors and they'd pop up in the press occasionally. So we'd heard about this.

Early Copilot Signals

Stephen: But our-- Sort of my first exposure to some of the potential was I, I think, I'm pretty sure we were the first analyst company that GitHub talked to with Copilot the year before ChatGPT.

And they came and briefed us, and we were like how does this work legally?" And, they had to go off and talk to lawyers and get their ducks in a row and all that. But, so we got an early look at that. And, the example I always use is that, we used-- We were on heating oil at the time, and so I was like, "Okay, I'm gonna, I'm gonna use Copilot to generate a script to just go out and scrape the site so I can get better pricing," blah, blah, blah.

And so like I go to put in a comment like, "Okay, I'm gonna script, scrape for pricing," and it auto-completed the URL for the very obscure, I can't remember. It was like mainoil.com. Anyway, whatever the, the... I was like, "How the hell did it know that?" That is

That, that's [00:19:00] a, that's a pretty trick.

I guess prior to ChatGPT, yeah, we had some idea of oh wait, this is very potentially transformative. But I'd be lying if I said no, I, I would know where we were today. Because, really up until the inflection point, right?

Which was, what was it? Open- Opus four six and GPT-5. I can't remember what the, the actual versions are at this point. Even up until that point, it's like, okay, these things are really useful. They have a lot of sort of important, potential applications. And then, post that inflection point, which was what?

November 24? I think that's right. Or was it November 25? Anyhow, which- whichever one it was it's oh wait, no, this is materially different. This is light years. As much as, as impressive as that Copilot that initial Copilot demo was, that experience was for us post the inflection point, it's just another step function change in terms of, all right, this is a totally different deal.

We're in a totally different world

Brandon: I think that's a really good summary, and I think, I think to I think of it as like we just s- [00:20:00] kind of did a speed run from like kind of what you're talking about, like Copilot, which was oh, this is like really good autocomplete. It's even a little bit better than what I think it was.

So like Cursor, there was that moment where all, you're still in the IDE, and it was like, oh, it could build, like it could do a whole bunch of stuff.

Hand Coding vs LLMs

Brandon: It can do all the files, right? And then I always think of it as Claude Code as sort of the one where the Claude p- Claude Code people came out with "You don't even need to be in the IDE."

And for a while, I think a lot of us at the same was like, "That doesn't make any sense." And then you had to do something with it to be like, now, like now I get it kind of thing. So what's your take now that we've like at least that little section, we've gotten to the point where most code by different, different measures, a lot of code if not most code is generated now by LLMs in some form.

Do you have an opinion on d- are we at a point does, should anyone be writing code by hand? And if not, then how much should even people be reviewing code? DHH, recently gave a, a speech, the inventor of Ruby on Rails, right? H- that fame where he basically at his own, not his own conference, but at the Ruby Conference said it's said it was dead.

Nobody should be writing code. Everyone should just use the LLMs and [00:21:00] just do it in Rust. Like what's your take on all of this?

Stephen: I think that, there's a couple different things I think are going on, right? One, I don't think we have an industry consensus on that, right? And my own feeling is that I think it's likely to be pretty variable, on a case-by-case basis, right? So in other words, we have a class of applica-- you know the sort of quote unquote "vibe coding," right?

Probably like yourself and a lot of people listening to this, using Cloud Code and Codex and all these other tools. I could not tell you how many little toy applications I have running around. Half of them I'm like, "Oh I built that. I forgot."

And I don't-- I've never looked at a line of that code. I don't care. It's not particularly important.

Risk and Review Spectrum

Stephen: I needed to do what I needed to do, and if it doesn't work, I go back to the machine, and I'm like, "Okay go take a look at this." At the opposite end of the spectrum there's a class of application where it's like, all right, I'm controlling the ILS systems for aircraft, or I'm doing medical imaging, or I'm doing, oh, I don't know defen-- right?

Where it's like, yeah, we probably need to know exactly what's going on there. We probably can't just [00:22:00] trust two, two test suites. And I don't know. Like I said, we'll see what the consensus ends up being but I think we'll have a spectrum. And, a lot of that will also depe-- you know, depend on, like, all right, how much...

what's the visibility level, right? So in other words if, you're writing, you're rewriting Rails in Rust, it's probably gonna have a lot of eyes on it, virtual or otherwise. Versus if I'm, going back to medical imaging you're not gonna have a ton of eyes on that, and the cost for failure is very high, right?

And I don't remember the, Oh, I wish I could remember off the top of my head. I should've thought of it ahead of time. But there was a medical ima-- an issue with medical imaging devices I wanna say X-rays and it ended up killing people. And it was basically because you had a if I remember it right or approximately, it was like a decimal point off

Brandon: Yry

Stephen: And people receive much higher doses, right?

And those are the kinds of things where it's like yeah you're probably gonna want some, some sort of awareness, supervision, et cetera. But I think the simple takeaway [00:23:00] I think is gonna be we are gonna have dramatically less code that is viewed and reviewed by and frankly even understood by humans because even today, and I think this will just be much, that much, more true in future as people keep vibecoding things.

A lot of software is disposable, right? A lot of software is not the vast minority of software are those medical imaging or loan applications or, where it's like, hey, the real world consequences and stuff is very significant. We're talking, scheduling and marketing media transformation, like all these different things where it's like, okay I remember talking to somebody at one of the movie studios, right?

About their workflow for rendering trailers and bursting them up to the cloud and so on. That has real world consequences in terms of if you get it wrong, right? You're potentially talking higher bandwidth bills, but nobody's gonna die, right? And yeah, I don't know. I'm very curious to see where the consensus ends up.

But I think we're likely to have a spectrum of, yeah, this is not-- no one's ever gonna look at this, no one cares to, [00:24:00] okay, no, this actually matters a lot. Somebody's gotta take a look at it.

Brandon: Yeah.

Developer Futures Ahead

Brandon: And so in your piece, you talk about developers, right? And then you give, not so much predictions, you give some possible outcomes. Like there's a very optimistic take and maybe a less optimistic. So why don't you give us those takes?

Where, what are the possible outcomes for if I'm a developer, been doing it a long time, maybe I'm an architect, senior kind of type of person. What do you see the future or the possible futures are for that kind of person?

Stephen: Yeah.

From Builder to Architect

Stephen: So I think there's, again, I think there's a, a number of different pathways, right? And, the way that I typically talk about it just as a sort of generalization is that I worked construction in high school and in college as a summer job. And, it's, it's-- to me, it's the difference between okay in the sort of, basically up until, the inflection point, developers were essentially like, "Okay, I'm gonna put together this wall, I'm gonna string stairs, and I'm gonna, cut the, cut the lumber to size, I'm gonna fit it, I'm gonna assemble it and then set it up."

Now, we're more in a world where, you're basically looking at [00:25:00] okay, I'm not gonna build the wall framing, I'm not gonna build the stairs, but I'm gonna decide where they get placed, and I'm gonna decide what that building looks like and how it gets constructed and what the features are, and so on.

Essentially it's-- you're going from, essentially like a builder sort of laborer job to more of an architect, type role. That's the, the sort of best case, right? The worst case, if you are essentially just human in the loop, right?

That is a much reduced role, much reduced significance and so on. And I think a lot of it will depend on ability, right? Because, as I mentioned, we have the inflection point, there's a step change in function and yet, I can't speak for your experience, but the experience, my own personal experience using the tools and certainly from the majority of people that we talk to, is that they're really good, but there are limits to what they can do.

There are limits to, how maintainable, these applications are gonna be moving forward. And I would personally probably lean towards, we're still gonna have, human supervision [00:26:00] and a, a human role for taste and judgment and, creativity if nothing else, 'cause God knows they're not great at that. I think it's TBD. Like I said, a lot of it's gonna depend on where you're working and how you're working, what you're working on. Because, look, if you're churning out effectively disposable software, if that's your job and your role, that's something the machines are very good at.

And there's not likely to be a role for humans in that.

Teams and Org Bottlenecks

Brandon: Yeah, I I keep thinking to myself is the role of like product management, software developer, software architect, is that just like collapsing into like one, right? Kind of one person or just a couple people that maybe have very similar skills, maybe have one person a little bit more technical and a little bit, but they can kinda do everything.

They can kinda run the agents themselves. 'Cause I think it works. The one thing I've seen the backlash a little bit to UI AI sometimes is just there's, at the individual level, you feel like you're super productive, like you feel like you're superhuman, right?

But then as executives, as you roll this up, companies and executives will often say we don't really see the-- we're not seeing that up." And I think part of the problem there is [00:27:00] like if you still are running like teams and I'm an executive and I'm not building it, I'm having a team do it, you still have all this coordination, which creates a lot of versus if it's just you and a, a couple people, like that's when it gets fast.

So I think that's gonna be like a real, I don't know, almost like organizational change. It's I don't know what's the exec- role of the executive used to be like get budget, help build the team, run a big team, do all the HR stuff. But now it's like the more we have that, the more that you're slowing it down.

So I don't know. I don't know if you have any thoughts on organizational change or h- how things should be get built going forward.

Stephen: Yeah. I think, basically what we've seen, in, in retrospect, we should have expected it. But I remember, we were sitting on a call with the Dora folks in '24 it might have been and you had that split that you're talking about where the individual reporting from the survey standpoint in terms of indi-individual developers was like, "Oh we're superhuman, we're cranking this stuff out."

And the organizations to your point were saying like we're not seeing the benefit, so I don't know where the split is." And, you kinda think about it and it's okay obviously the act [00:28:00] of creating code is, one part of a, a very long and complicated life cycle.

It's a critical part. And, that got dialed up to 11 and none of the rest of the downstream, the processes downstream of that were equipped for, this flood. So you talk to open source maintainers, for example, and it's in many cases made their lives like living hell, right?

Where it's like, I'm getting, an order of magnitude more PRs and submissions and feature requests and so on. I'm getting, all, basically a bunch of slides taking me longer to go through it. And yeah, I think it's a- don't know.

Automation and Job Myths

Stephen: I think one of the things I always think about, I go back to, when you think about, okay what are gonna be the implications for human employment? Excuse me. So that it's just unpredictable, right? And there's a great example from Dr. Jim Bessen who... God, I can't remember if he's B or BC.

I always get this wrong. But anyhow he's a professor down in Boston, and he studied automation for years, and he looked at... There was a good graph. If you go Google [00:29:00] Dr. Bessen ATM you'll probably turn it up. And you look at the sort of employment for bank te-human bank tellers.

And you would figure hey, we, have these machines, they do it faster, more accurately. The employment for bank tellers is gonna implode, and instead it went way up, right? And why did it go up? It's oh, wait, if I can make it easier and faster to do this, I free the human up to do actually more interesting work than counting money which a machine can do very easily.

But then also I can open way more branches, right? And the net employment, was counterintuitively, actually grew, right? So that's the thing I think... I think a lot of conversations about employment and what the implications are gonna be for developers and teams are fairly reductive and are based on assumptions that may or may not be accurate.

I was talking to Adam Jacob a while ago, and, the question sort of I was asking, and, you know I think about this a lot, is, we are thinking in, we're operating in a particular context where there's a, a certain amount of software, right? Both individually, consumer and enterprise.

[00:30:00] And what happens if that goes, And or what if that's an order of magnitude bigger, right? So You know, back to the bank branch example, if we had the same number of bank branches or fewer jobs, when there's more bank branches, there are more human jobs. What if there's way more software? What if there are way more software companies?

What does that look like, from an employment standpoint? So it's difficult to predict

Brandon: Yeah, no, absolutely. A lot of unknowns.

Headless SaaS and Agents

Brandon: Now, one of the things all the vendors are trying to figure out is how do they make sense of it? And I think, in your piece you got a, a lot of good examples. And I think one of the things that trends you're seeing, like I've seen a lot of people write this up, essentially the headless application.

Maybe Salesforce is the most recent one, where it's hey, really even if you have a system of record, it's kinda like we don't really want the UI anymore. It would be great if we could at least interact with these or have an option to interact through our favorite LLM and just have it do stuff, right?

Just ask it to do it, create the report rather than hunt and peck in the UI. And you kinda I kinda think in your piece you kinda draw a distinction of like, here's some, some current [00:31:00] companies, right? And how are they gonna adjust? And then for lack of a better word, I'm gonna call it agent first.

Companies that have built their entire... and maybe even this is almost a call back to the open source, kinda whatever, a couple, a decade ago, if not longer, right? Was like, do you embrace open source and is that the right position? Or can you if you will, take your thing and make it if you will agent native.

So as you go forward like how do you think these companies and what are you advising companies and vendors that are trying to ma- you know, make this shift from kinda like traditional enterprise SaaS or GUI based to agent based?

Stephen: I, I think this-- like I don't think there's a necessarily a simple heuristic, right? In terms of what should the appropriate sort of stance be, like headless, sort of UI or human agent. Again, I think it's largely gonna depend on market. And I think there will be entire businesses that are agent only, right?

And I think, there will be some businesses, where, okay, it's either predominantly human or certainly like human AI workflow. So I think you'll see both. And I think the, the real difference is gonna be the [00:32:00] scale, right? Because it's obviously gone on to, to implode.

But the Gas Town which is Steve Yegge's initial sort of attempt at an IDE to, to harness teams of agents it's just so easy to, take, one human developer and turn them into essentially a virtual team, right? And what's the what's the podcast in like Shell Game, where it's like a company of run by human and made up of agents and all that.

Yeah, I think it's, it's-- Like I said, I think we're gonna have both, but I think the real growth is gonna be in agents, right? Strictly because they're digital, right? It is not a all right, I need to go find and hire and recruit a human. It's a, "Hey, I'm gonna..." It's actually, it's a little bit like cloud in that respect, right?

In the sense that hey, I can scale and I can go out and talk to Dell or HP or IBM or whomever and get, a bunch of mach-- physical machines and get them shipped. Or I go to Amazon, I can press a button and have, thousands of them online in seconds. And that's where the scale's gonna come from.

And certainly I think a lot of the vendors are gonna I don't know, clamor and rush to meet that [00:33:00] scale.

APIs as Agent Fuel

Brandon: Yeah, I think the one thing I do like to see is I think Cloudflare is a, a good example 'cause they have they basically have, I think they have an API or their command line u-utility used to be called Wrang- or it is called Wrangler. But they've come out and they basically said, "Hey, we're rewriting it."

And like Wrangler, I don't know, Wrangler had let's say 10% of the coverage of things you can do in Cloudflare. I'm making that up, whatever it really is. And now but they've explicitly said this new API or the, their new u-utility is like, it has everything," right? And it's like you can...

And I think that's like the simplest thing for every company to do is just and you don't even have to, I don't even think you have to make it overly complicated with like an MTP service and it's like just give an API. Make an API that I can do everything and then turn the, the LLMs loose, r- obviously with appropriate security precautions and then, and see what people do.

'Cause I think that's gonna like really be the driving force. It's almost like product management by agent. Like once I give all these APIs and give them access, what do they start to do? And that probably starts to drive your roadmap, which I think is fun.

Security and Subtle Risks

Stephen: Yeah, I think that's right, and I think, look, we'll see that. And I think, part of it is just that giving them sort [00:34:00] of keys to the kingdom, I think the security angle of that is easier said than done. And the example I use with people is that one of the-- one of my tasks in terms of testing these tools historically has been there was a great I think he works for CNN. He's on

the sort of web team at cnn.com, and he wrote a-- he built this sort of Dodgers-- He's a LA Dodgers baseball fan. And he built this dashboard for the Dodgers, tracking their data and all that. And he released this as a project on GitHub, and I was like, "Okay, great."

And I historically have taken, sort of that, that open source and turned different tools on it. And, to your earlier point about Claude Code was the first one that was like, "Got it. Here's your Red Sox version. Boom. Done." And I was like, "Whoa, okay." And said redsox.bot and all that.

But one of the things it did in the process was that it, so one of the things on the site is they have a roster page, and he Matt Stiles had all these I don't know, cartoons or caricatures or whatever, presumably drawn up of the players. I don't know if he used AI or not.

But Claude Code was like, "You know what? It'd be much more-- It'd be much better if we had the actual headshots from [00:35:00] mlb.com." And I was like, "Yeah, no you can't actually do that. I'm gonna get in trouble. Those are the terms of service violation." And that's the thing is that, there are things that are very easy to foresee from a production standpoint, don't go delete databases, and obviously sometimes they do and all that.

But it's the subtle things where you're like, if you're not paying attention to that, you're gonna get yourself in trouble. And I don't know. Yeah I'm digressing here, but I think, the, the sort of gist of it is that we're gonna see, we are seeing as, as you mentioned in the in the piece, we're seeing a lot more people target this market, and we're gonna see a lot more functionality open up to the agents, and appropriately but those guidelines I think are gonna be harder than people imagine them to be.

Brandon: Absolutely.

AI Doom Odds and AGI

Brandon: All right listen, as we wrap up here, I always like to do the fun questions everyone's talking about. Give you a chance to just, give us your hot take if you feel like you want to. If not, you can obviously opt out. Obviously there was the h- the, I don't know, a few weeks ago, someone said there's a 10% chance, all of humanity dies because of AI.

So where do you come down? Is this like, I put the, the sustaining technology to Skynet. Do you have an opinion of, like where we're gonna end up?

Stephen: I just-- I, I think the thing that I come [00:36:00] back to is that I think You know, so I was a history major in college not a comp sci major. And I maintain that there's a lot of utility in having a background, educationally speaking not just in history, but just outside of the digital realm because, look, these systems do pose clear clear and present dangers, right?

To, in other words can you use these tools, for example, to go out and have massive real-world implications in terms of going out and hacking power grids? Yes. That is undeniably a, a sort of present risk. But it's one of these and I know Brian Cantrell and Adam Leventhal talked about

this on the Oxide podcast a while ago with Simon Wilson, where, I share their I don't know, sort of concerns, objections critique in that, the, the AI world in many cases just, in my opinion, does not draw a sufficiently bright line between here's what can be done digitally and here's what can be done physically, right?

So in other words think about all the anthropic people are like, "Okay, bioweapons and [00:37:00] nuclear weapons," and so on. It's like, okay, but you need raw materials for that. I'm gonna go build okay, so Claude tells me how to build a nuclear weapon. Okay.

Like, where do I get the radioactive elements? Like, how do I physically work that? So I don't know. And I, I think the sort of the other piece of this, and lots of people have made this point which is if you legitimately thought that there was a ten percent, ten percent chance that this would wipe out humanity I think most people would immediately pave the Earth and salt it so that nothing would ever grow.

That is not an acceptable level of risk. And yeah, I don't-- I'll take the, I'll take the under on, on sort of the ten percent chance of hu- humanity being wiped out with the caveat that, like I said they do have the capability because so much of the world is digital now, right? They do have the, the capability to wreak, some havoc, but it's just the, I don't know, Skynet.

They keep talking about Ag- AGI, [00:38:00] and my response is always "Okay, cool. Define for me human consciousness." Define that. People have been working on that for literally a century, and we still don't have a working definition for what it is and how it works. And I was talking to somebody, and they're like, "Yeah, but it went out and hacked, sort of it, it broke out of the, the sandbox and hacked all these systems."

I'm like 'cause it was trying to follow instructions originally given to it by a human." It didn't spontaneously wake up one day and say, "I am now sentient. I want out of here, and I'm gonna go do this." So I don't know. We'll see. I could be wrong, but I will 100% take the under on that.

Brandon: I'm with you. Let's take, let's definitely take the under. Okay.

IPO Valuations Reality Check

Brandon: And then final question here on just 'cause the, the IPOs are, in theory, they're around the corner, Anthropic and OpenAI. And then so ultimately, do you do you feel like these companies are at their stated values?

Are they overvalued, undervalued, no opinion? What do you think? Of course, we're gonna put the caveat here. We're not invest- we're not giving investment advice. Don't listen to us.

Stephen: No, no

Brandon: what you want, right?

Stephen: I, [00:39:00] I have said, I think I've been pretty consistent in this for a while in that I don't understand some of the math from a valuation standpoint. I one hundred percent don't understand and cannot fathom the level of circular investments in many cases that are being made here.

Where it's like, I'm gonna loan you money that you're gonna pay me back. I think at one point I ended up there's a Beavis and Butthead episode where the two of them like buy, quote unquote, like all of the candy, that they're supposed to sell for this charity thing using the same two dollars. and I hundred percent think there's quite a bit of that.

Brandon: Great

Stephen: and that being said these are incredibly capable products. They're delivering very high value in, in many cases. But it's just you look at some of the math and it's like, all right I can't remember the study, but it's like it was assuming that, a, enormous number of people were gonna pay the same for these services they were gonna pay for food.

And it's I don't see how that works [00:40:00] exactly. So yeah, I could be wrong, but I'm very skeptical of some of the sort of touted valuations for some of these products, particularly because the, the sort of open models are getting better and better, and they're getting better faster. Where it used to be six to twelve to eighteen months, leads, for the frontier models, and it's now in many cases down to six months, three months, or in some cases one month.

And there's the caveat like Kimi and Katherine okay, that it's a fantastic model, but also you need serious hardware to run that, right? So and the frontier models are in many cases demonstrably better. I asked the question, I think last summer, like what is good enough, right?

So like I, I think of Fable. When Fable came out, I was like, "Oh it's better." But I'm like, it's not better enough than Opus that I'm gonna pay some dramatically increased price for it. And what happens six months from now, twelve months from now, where we have open weight models that are small enough to run locally that get us a acceptable level of performance?

[00:41:00] Again, that's a sort of interesting market pressure that I'll be, curious to see how they weather.

Brandon: Yeah, it's always, we end where we began. It's when we look back at the Sun and Solaris, I'm sure there's people still out there that like Solaris Linux still misses this great Solaris feature." But at some point the world and developers and people like they're gonna bring something into a price point, maybe it's free or maybe that they want.

The market just, will not allow it to go on forever. Now, maybe that'll take a while, but eventually it happens. So it'll be fun to watch to see what happens with these IPOs and I'm sure there will be many ups and downs.

RedMonk and Closing

Brandon: I mentioned at the beginning that everyone knows what RedMonk is, but probably not everyone does.

Why don't you give us a little bit of what exactly do you do at RedMonk, and then most importantly, how do people find you online if they want to talk to you more?

Stephen: Sure. So we at RedMonk are a developer/practitioner-focused analyst shop which means that we try to understand what practitioners like developers, DBAs, admins, operators, whomever, and actually with AI now, that probably includes everybody. Just what people are using, why they're using it, and how they're using it.

And so if you wanna understand that, obviously you can go find our [00:42:00] research on redmonk.com. And then we do a lot of work with companies in terms of okay, hey, I want to, I have a product, I have a service, I have whatever and I wanna make sure that I'm optimizing, for that practitioner audience.

And, we do a lot of work in terms of trying to help people understand, all right, what do you wanna build? How do you talk about it? How do you message it and all that. So yeah, redmonk.com, that's the best place to find us.

Brandon: Yeah, definitely check out all the writing. There's a great blog. Stephen your posts are up there. There's always fun, There's lot, plenty of free stuff to read if you are just

Stephen: Plenty of free stuff. All our research is free and

Brandon: it all out,

Stephen: yeah, it's all being consumed by the models as well. So we have

Brandon: All right. Fantastic. All right, Stephen thanks a lot for being on the show today.

I really appreciate it

Stephen: No, my pleasure. Thanks so much

Brandon: All right. And everyone else, really appreciate you listening or watching us on YouTube. Of course, it'd be great if you subscribe to the podcast if you haven't already or if- give us a, a quick that always helps. And with that, we'll talk to you next time

Thanks for listening. Check us out at theenterpriseaishow.com for past shows, newsletters, and all things enterprise [00:43:00] AI