

EPISODE 998

[INTRODUCTION]

[00:00:00] JM: The Changelog is a podcast about the world of open source. As open source has become closely tied with the entire software development lifecycle, the Changelog has expanded its coverage to the broader software industry. Since starting the podcast 10 years ago, Adam Stacoviak and Jerod Santo have become full-time podcasters and they've started several other podcasts within the Changelog network including Go Time, JS Party and Practical AI.

Throughout all of their shows, there's a consistent theme of technical, entertaining conversations about software. In the last decade, so much has changed within open source. GitHub became the de facto social network for open source. Kubernetes created a widely used platform for distributed systems. React has given frontend developers a component system to consolidate around.

Adam and Jerod return to the show to discuss their perspective on the past and future of open source and their learnings from interviewing influential software professionals for 10 years.

[SPONSOR MESSAGE]

[00:01:10] JM: Today's episode is sponsored by Datadog, a modern, full-stack monitoring platform for cloud infrastructure, applications, logs and metrics all in one place. From their recent report on serverless adaption and trends, Datadog found half of their customer base using EC2s have now adapted AWS Lambda. They've examined real-world serverless usage by thousands of companies running millions of distinct serverless functions and found half of Lambda implications run for less than 800 milliseconds.

You can easily monitor all your serverless functions in one place and generate serverless metrics straight from Datadog. Check it our yourself by signing up for a free 14-day trial and get a free t-shirt at softwareengineeringdaily.com/datadogtshirt. That's

softwareengineeringdaily.com/datadogtshirt for your free t-shirt and 14-day trial from Datadog.
Go to softwareengineeringdaily.com/datadogtshirt.

Thank you, Datadog.

[INTERVIEW]

[00:02:23] **JM:** Adam and Jerod, welcome back to Software Engineering Daily.

[00:02:25] **AS:** What's up? What's up?

[00:02:26] **JS:** Hey, thanks for having us.

[00:02:28] **JM:** You guys started Changelog 10 years ago, and I think we could start by just taking a little bit of a retrospective. What are the most significant changes to the way that open source software is built over the last 10 years?

[00:02:42] **AS:** I say it got more progressive rather than change, right? It started out sort of like back in 2001 when GitHub launched, this new era of collaborative. Before you stepped in email diffs and all these weird things, like this weird collaboration processes, it only sort of propelled from there. We have more and more popularity around projects, more and more collaboration as a community efforts have gotten deeper. Foundations, profit obviously. There's a lot of money in open source now it seems.

[00:03:08] **JS:** Yeah. It was like the sea changed happen probably right before the podcast began, which was with the public launch of GitHub and the beginning of the GitHub era. That dramatically changed the way that open source was conducted mostly because it brought the source code to the forefront and really the collaboration and the social aspect of coding, which is so much of it. Especially when you're a hobbyist or an open source maintainer, it brought that right up in the forefront. They pull the request flow, which is so popular now, was really coming out of that, and it changed many things.

Then from then – Yeah, I agree with you, Adam. It's been progressive since then there's been a lot of minded and a lot of corporations coming into it and there's been a lot of individuals suffering under the thumb of open source and trying to find ways of making some money or getting support.

[00:03:57] AS: Yeah? What about this one? Where you have a really crucial open source project that has a vulnerability that has actually affected the cloud or clouds or large complications and caused outages or big issues? That's a new thing in this decade.

[00:04:13] JS: I mean, I guess this probably happened before, but it seems more often. Everything from Heartbleed on to recent brand-new news in the Rust world which I believe is actix. Somebody who stepped away from an unsafe – We could say it on your show, Jeff, shit storm.

[00:04:32] JS: What do you mean they stepped away from an unsafe ship storm? What's that even mean?

[00:04:34] JM: What are you referring to?

[00:04:35] AS: Hey, man. It's like brand-new bleeding edge news. The repo on GitHub is actix, A-C-T-I-X. Then the actix web. I believe it's a web framework for the Rust world. I'm not sure the context, but I know Steve Klabnik talked about it and he's really sad about it. There're some details there that I was trying to read into. This is literally bleeding news, like 5 hours ago, bleeding edge news. I guess –

[00:04:58] JM: Well, and bleeding news.

[00:05:00] JS: There's some bleeding going on.

[00:05:01] AS: Yeah.

[00:05:02] JM: Some people are suffering from it. What about the importance of corporate backers? Open source was historically this thing that was – The earlier days of Linux was very

decentralized, non-corporatized. Overtime it's gotten increasingly corporatized to what you have today with just the rise of open core companies and the cloud co-opting open source projects and productizing open source projects. What's the relationship between the corporate backer and open source projects today?

[00:05:40] JS: I would say the relationship right now is that the corporations writ large are not really backing them in any sort of financial way. Now there are exceptions to that, and we see movement in that area. But I think if we're just speaking about corporations and the relationship to open source over the last couple of years, I think they've been kind of on the take on the receiving end of the goodness and less on the giving end.

We're starting to see changes there. We're starting to see people who are in the open source maintainer world demand and more support and we're starting to see value exchange and saying, "It's difficult to just ask a corporation for donations, but you can provide a value exchange around as project," and that is something that's a more feasible line item for them. It's giving them ways to give back and still receive. It's a give-and-take. I think, historically, the open source maintainers have been more on the give and the corporation has been more on the take, but it's starting to change.

[00:06:40] AS: The primary ways they're giving is in which ways though? Just primarily financial?

[00:06:44] JS: I think the primary way corporations are giving right now is by allowing their employees to work on open source on business time. That's probably the primary support.

[00:06:53] JM: I was more curious about the corporate stewards of the open source projects, the companies that are refunding the open source projects. I was interested to get into a discussion of if you guys had any perspective on the licensing changes and how these database companies are feeling like they need to insulate themselves from AWS through licensing changes.

[00:07:18] JS: Okay. Yeah, that's the thing about open source, is it's such a big thing and it's hard to drill in without specific. I was coming from the maintainer's perspective of like open SSL,

for example. It's this massively used SSL library that was maintained by a couple of people who were not funded, whatsoever. Hence, hardly back in the day and really an eye-opening for us as a community, like, "Wow. Some of our core infrastructure is completely under maintained."

From the perspective of like Facebook, Google, these companies coming out with corporate-backed open source projects such as Kubernetes, such as React. I think there's a lot of strategic reasons why they would do that and there's a lot of value there for us as individuals, because think about the amount of labor put into React by Facebook, right? The investment there is massive.

In that case, the indie devs, the small businesses, the open source maintainer, we all benefit from that because I could not build a React on my own and they want to have it focused around – Well, specifically like Kubernetes. I remember we had the Kubernetes folks on the Changelog and we asked them why open source? It seems like something you could monetize – Or it seems like a secret sauce. Why open source that?

Their response in summary was like cloud orchestration was going to be a thing and it's something that everybody does. We wanted it to be a thing that we were drivers of. We want to be at the center of that ecosystem. I think there's lots of reasons why corporations go open source, and that's one of them.

[00:08:59] JM: Do you guys have a sense for how Kubernetes has affected the competitive dynamics of the cloud providers? Pre- Kubernetes, it looked like AWS was just going to run away with the entire cloud business, and Kubernetes has really opened up an opportunity for Google to have an in – I suppose, Microsoft had an in even despite Kubernetes just through their integrations or enterprise sales channels or whatever. But do you have a sense for how Kubernetes has changed the competitive dynamics of the cloud?

[00:09:31] AS: It's flattened it, right? If AWS was sort of leading that front in a lot of ways, they had the secret sauce, which was in a lot of cases getting your stuff there and managing orchestration. With the open source of Kubernetes and everybody being able to use it, to also add on to that, the CNCF and the efforts being done there to sort of bring everyone to the same

table, I think it goes back to what Jerod was saying around React where you have this massive investment from Facebook, and that just can't be quantified. It's very difficult to quantify it.

We take the same lens and apply to Kubernetes where you see all these clouds pouring into the same orchestration infrastructure and community. It levels the playing field, but it also makes the software we use to make that happen so much better. The footprint is – The visibility into what Kubernetes is and where it's going is seen by all and everyone's welcome to the table proverbially. To some degree, there's an invitation there for everybody.

[00:10:30] JM: Do you guys have any perspective on whether AWS has taken too much oxygen out of the room? There is Redis Labs, to Elastic, to MongoDB. There is this perspective that AWS has taken an unfair amount of value without contributing back to open source repos. This perspective kind of assumes some norms around open source. The idea that if you get some significant amount of value out of an open source ecosystem, by monetizing it, you have some obligation of giving back. Is there an undue amount of value being taken by AWS from the open source ecosystem?

[00:11:19] JS: I don't know who is the arbiter of what is undue value. I think it's a case by case basis. I think if you ask the creators of – Or I say the business, Redis Labs. I think if you ask Elastic, then the answer from their perspective is, yes. I think if you ask Amazon from their respect, I think the answer is probably no.

As an observer, I land somewhere in the middle. I can kind of see Amazon's perspective. I also see where we had Adam Jacob, a great conversation with Adam Jacob on the show where –

[00:11:48] JM: I enjoyed that episode a lot actually.

[00:11:49] JS: Thank you. His perspective is I think unique, because then Adam had Sentry's David Kramer on and asked him about Adam Jacob's perspective, which I'll share here so we're all in a loop, but he disagreed with it. So I think there're people on both sides of this.

Adam Jacob's perspective on this particular subject is that what in AWS does, when it adapts your open source project that you have a business built around is it validates it in a way. Now, if

you have paying customers and stuff, it's probably already validated, but it does do that. It gives Amazon very vested interest in that thing remaining and then being able to provide that service for a long time.

Then it also dramatically increases the top of your sales funnel. As their platform is so large, it dwarfs – Their customer base is so large in comparison the most other businesses, especially open source-focused businesses, that it opens up a huge sales funnel for those businesses. People that never would've come across their product and now they get Amazon's default install of Mongo, or of Elastic, and then it allows you to then compete directly with that default, which they're never going to really have much of a value add.

I mean, Amazon when they add new services, it's basically like the default service with SLA's, and interface, and an API. But his argument is like that's very easy for a business to compete against, because, Amazon, their interest is like how many services can we add versus can we make this like the most awesome feature-full version?

That's his take on it, and David Kramer's says, of Sentry, he's like, "Yeah. I don't care. I don't want them to run my software basically for free and resell it." It depends on how you ask. What do you think, Adam?

[00:13:39] AS: I think it's a hard one to answer. Honestly, I agree with you. It depends on who you ask. I'd like to also know the arbiter of unfair value or undue value. I personally don't follow AWS's interactions and open source well enough to really know what they have done or what they haven't done.

Because open source is the way it is, I suppose that's sort of a social contract, not an open source license that requires you to participate, right? You can freely use it and freely do it and now you have different licensing options, like business source licensing, or I think it's called zero license or something like that. There's a couple of different options now that sort of enable you to do what Sentry has, which is done eventually open source, which is they put their stuff out there. It is open source, but it has a time, a ticking timer basically till actually becomes open source. It will be eventually open source.

That opens up some unique options for Sentry's of the world to still be in this open source world, but then you get this sort of backlash of like, "Well, is it really open source?" Then the people that generally sort of fight that battle and sometimes literally go to legal wars with it is the OSI. Kind of protecting the word open source. It may be in spirit open source, but calling it open source is the thing I think people begin to scrutinize.

[00:14:58] JS: I think your point, the word you chose there was norms, Jeff, and I think that that is astute because Amazon is completely inside their rights and inside all legalities to do exactly what they're doing. The question really is about like moral norms. Then you're talking about the norms of a community that's global, spans ages, spans business sizes from a single developer all the way out to Microsoft of 55,000 or however many people that work for them. What are the norms of this community? It's very difficult to nail those down because everybody has their own ideas about what those are.

[00:15:33] JM: You know it's funny is I feel like there's a sensibility in the developer community that as soon as anything gets too profitable, we just reverse engineer some reason to criticize them, and it's funny because you would think like, "Okay. Look, Linux took over. Linux is just the operating system that everybody use. Linux is the best."

We never say Linux has too much market share. It's just an open source thing. But as soon as somebody is making a preponderance amount of money off of something, then we instinctively look for a reason to find some criticism of them. Maybe it's just the source of human nature. I don't know.

[00:16:23] AS: Don't be too successful, right? Don't make too much money.

[00:16:27] JM: Basically.

[00:16:27] AS: Don't go too far, because if so, there is some backlash. I think it comes from equality, right? You want to have some – I think it's about being able to access. One something gets so big, so profitable, take Amazon for example. If Amazon launches a business, and this is sort of in the capitalistic side of things. If they launch in a business in a world, say, they want to acquire. Well, the grocery market of Wholefoods gets very dynamically different. If they want to

release a particular style of business, every other business in that market will then be devalued in many ways because Amazon is such a Goliath.

When you have that kind of power, it's specific to Amazon and then by way of AWS, then is an unfair advantage to some degree. That doesn't mean it's not earned and they didn't get there by moral means even. They've done it in a way that hasn't sort of like screwed over people. They're just so big now and people – You don't want it to be that big. You want to be that profitable, because it's a large amount of power in one hand.

[00:17:31] JM: One subject I wanted to ask you guys about was the two businesses of GitHub and GitLab. GitHub obviously was core to the rise of open source, and then GitLab came out and started off in some ways looking somewhat similar to GitHub. It looked like kind of just basically an open source version of GitHub. But since then has gone in a totally divergent direction. How do you see the two businesses of GitHub and GitLab developing in their divergent business lines over time?

[00:18:08] JS: I think that GitHub is most well-known for open source. It's almost as if if you're in the world of open source – We just had this conversation to some degree. If you're in the world of open source, you're almost expected to be on GitHub. Not a self-hosted Git or GitLab, for example. That's where "open source" happens. Not my true opinion where a lot of it happens, but I think it happens in a lot of different places, but that's where the masses are, the critical mass at least.

GitHub has actually done some things in enterprise and done some things in other areas, but I don't know their exact revenue and where their revenue numbers come from in terms of a pie chart, but I would venture to say they have a lot of paying individual customers, a lot of paying corporations using it to host their open source projects and it's almost as if you want to be there so that you can have open source and close source in the same bucket. Might as well not even have a GitLab and a GitHub, why would you host in two different places? Maybe some people do. I don't know. Whereas GitLab seems to be more focused on the enterprise, enterprise customer base and that kind of thing. They've done very well there.

Their businesses are a little different in that front. They both do code hosting, but I wouldn't even say that GitLab has gone even further to say they want to provide an entire ecosystem for the developer and they've transcended this code hosting idea, right? They've got CI tools and various tools in their tool bag that make an entire developer workflow from ideation to shipping a thing in a collaborative way. Whereas GitHub has been focused on inching towards that in some ways with issues or different projects and stuff like that, but they've kind of – The cream of the crop has been the fact they've been the open source way and they profit off of that and build their business upon that.

[SPONSOR MESSAGE]

[00:19:55] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to softwareengineeringdaily.com/g2i to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to softwareengineeringdaily.com/g2i to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[INTERVIEW CONTINUED]

[00:21:44] JM: Architecturally, I think GitLab is also interesting as an open source monolithic system that does a lot of things. You have GitLab CI and GitLab static analysis, monitoring, all these other things that are bundled with GitLab. As an open source project, that's a pretty strong divergence from the UNIX philosophy, the whole idea of just do one thing. I'd say that the vast majority of open source projects in the past are something that's narrow. Some narrow NPM library or narrow database system, queuing system. You have this occasional big monolithic open source system like a Linux, like a GitLab, or a WordPress, or dev.2.

Is there something we can learn from that? The idea that there is this occasional giant monolith that comes out that is successful? What's the divergent breakdown? what makes a giant monolithic open source system successful?

[00:22:55] JS: Well, it's integration. It goes back to classic modular versus integrated products. Over the last 15 years or so, I think Apple has proven over and over again the integrated products tend to be better products in many ways. Of course, there are drawbacks as well. It's a tradeoff.

I mean, are you speaking architecturally? Because I think if you had like GitLab engineers on the show, I think they'll probably argue that they're using the UNIX philosophy and so far as they have these different architectural sections. I think maybe they would take umbrage of being called monolithic. But I think integrated is maybe a word they like better, because there are separate modules that GitLab has and I think they're relatively decoupled. You can plug and play different aspects of GitLab. It's an integrated product, but architecturally I think that they don't have like a monolith necessarily. That means that I take your point and I think that it's nice having everything in one place. It's nice having everything flow in and out of the same system and not having to glue together things ourselves.

Now, as developers, we also sometimes like to just, "Well, thank you. I'll just glue things together myself and I'll go from there." There's customers for that style, but I think GitLab is proving it and I think other companies such as Apple has proved that integrated solutions are

really, really nice because they are seamless in many places where your modular systems have seen.

[00:24:23] **JM:** WordPress, you guys both spent time in the WordPress ecosystem, right?

[00:24:27] **JS:** Yeah.

[00:24:28] **JM:** Why is WordPress so persistent? Why is that community seemingly indestructible when it comes to CMS?

[00:24:38] **JS:** That's a really good question. I wonder that myself sometimes. Why is WordPress not just still around, but is still dominating in the –

[00:24:46] **JM:** Nothing's come close, right? There's nothing that has even come close to competing –

[00:24:51] **AS:** Apples to apples. The closest thing might be something like Squarespace, which is not an apples to apples scenario, because it's not open source and it's not installable to platforms. There're a lot of people who use Squarespace to do that, like a CMS or a website or an easy website with a theme.

[00:25:06] **JS:** One thing I think WordPress pretty well early on was they moved beyond blogging and they allowed – Just with the pages feature and with the customization features, they moved from something which many people started at like free open source blogging platform and they allowed you to create non-blog websites early on.

Now that's commonplace today. That was where a lot of people move from like I use this as my blogging system to like, "Now, I need another website for my business. Oh! I'm used to WordPress. I know it. I can set it up with pages. It doesn't have to look like a blog. By the way, I can have a newsfeed, which technically is a blog. That's great."

I think that was one aspect of it, was it was malleable early on and didn't have to be blogish. It was CMS. It was more of a general purpose CMS maybe earlier than other platforms. I also

think WordPress really developed a rich ecosystem of plug-ins and a marketplace for those plug-ins. That's actually one of the things that bugs me now when I go back to it, is that all the plug-ins are like monetized and it's like the freemium models. Like inside your WordPress plug-ins, you have indie-devs either scratching out or having a nice living inside of this WordPress plug-in directory and finding ways – Which is cool as an end user and like, "Oh!" I remember the good old days when everything was just free and open source and now there's money involved and so like it feels icky. A lot of people started building businesses on top of WordPress, and once you get that there, I mean there's a network effect.

[00:26:38] AS: the first paying gig I ever had as a web developer was building out somebody's WordPress website for their business.

[00:26:45] JS: What year was that?

[00:26:46] AS: 2004. Yeah. Yeah, it's a long time ago. I'll add to what Jerod said. I think that the same thing. I think it's one part leadership of WordPress and one part software enablement. You've had this open source software project out there, pretty accessible. PHP was well-accessible. LAMP stacks were easy to stack up. The barriers to entry were very little at that time. FTP and moving things that way was a very commonplace. It was the only way at that point – Not the only way. [inaudible 00:27:18] come true.

I would say it's one part leadership and one part that, but you get this ecosystem of plug-ins freely available. You can easily trust going to WordPress and saying, "If I hit a hurdle, there's something out there that gets me over that hurdle. If I haven't developed it myself, which I can, if I can learn how to write plug-ins. Otherwise, there are plug-ins out there that are tried and true that people recommend. They have a marketplace for it.

Matt Mullenweg has also been a great advocate for WordPress and a great business leader, a great citizen of the community as well. He has a very good outlook. I think it's one part the enablement that the software has provided the users of it and then also this developer ecosystem. But then also, to not discount, Matt Mullenweg. I mean, the names are synonymous, really. WordPress, Matt kind of go together.

[00:28:01] JM: Yeah. Shifting from open source specifically to a conversation we touched on a little bit in the last time I was on your show, the idea of how programmers should think about their career, think about their work-life. We talked about GitLab a little bit. GitLab has pioneered a particular type of working. They have this transparency model. They have this large-scale remote workforce. They do a lot of things that are somewhat heretical or would seem totally heretical 10 years ago.

Today, it seems slightly less heretical, still somewhat unusual. What can be learned from the GitLab management model?

[00:28:48] AS: You mean about how it's very open?

[00:28:50] JM: Well, from the transparency to the remote workforce, to just – Well, I guess those two –

[00:28:56] AS: I think it's the way. I mean, you're going to see – We already see this. We already see people – I mean, I work from home right now. I guess that's a little different I work for GitLab, but there's a lot of people I know in the software world that work from home or have a varied – They might even live in a 30 minutes or a short train ride away from the office and they still work from home the majority of the time, because it's a thing.

Now, in terms of the openness, I think that's the way. I think you see more and more businesses having to sort of like set an expectation of openness and inclusiveness, because that's sort of what our community has demand. It's what's desired from the community. To be a business and go against that and not play with that, I think Sid has been very forthcoming and very future-looking when it comes to that because he said, "I want to create a company that people want to work at." I think that's how you do it, is you attract them through inclusive and diversity and openness.

[00:29:51] JM: How transparent should it be? How far should the transparency go? Should we have all emails be public? Should everyone's compensation be known? How do we answer those questions of transparency?

[00:30:05] **AS:** That's slightly above my pay grade.

[00:30:07] **JM:** We don't even know what your pay grade is, because —

[00:30:12] **AS:** Well, basic math. You can go to our website and do some basic math and figure out our pay grade. But I don't know, honestly. I mean, I haven't studied the — I would say a data. I haven't looked at the deep data on this subject, but my instinct is like there's — When it comes to privacy, you don't know you want to be private until it's time to be private about something. You want your privacy when it's like, "Okay. Now my privacy is invaded."

I think in those regards, it's the that you don't want inequality in pay, but at the same time, as a business, you don't want to be forced to pay people, I guess, more than they're worth or more than the value of their position because the whole point of a business is to profit and to grow. That's how things work.

Now, there's ways to do that that are ethical and ways to do that or say slightly less ethical, but I don't think publishing everybody's paychecks or salaries is wise either, because there's — Aside from coworkers looking at it, you also have creditors and other people who get that same information or could get that same information. It's just internally. I don't know. I just don't know. It's difficult. That's a difficult one answer and I am so glad I'm not in that position.

[00:31:28] **JS:** I think what companies like GitLab have proven is that other models can work and that we don't have to do business the same way we've been doing business for the last N years, and we don't really know what's better. Experimentation is a good. A completely open, everything is published company needs to exist before we can know whether or not that's a good idea or a bad idea. Let history decide, right? Let the market decide if a company wants to come out and set a policy of publishing all employee salaries. Does GitLab do that? I don't know exactly how open they are.

[00:32:06] **JM:** I don't think so. Buffer did that. Buffer —

[00:32:09] **JS:** Yeah, Buffer did that. Like a radical transparency, I think people have called Buffer style. Well, do people want to go work for Buffer? I mean, because people will decide like,

"Well, I'm not going to work there because I don't want my salary published," or "That's great. Now I can see what everybody's making and I can negotiate better. I would like to have everybody salaries published." I think over time the market deices, I think we need more experimentation, I think.

What one thing that seems true so far is that remote work can work and that people like it. I mean, there's challenges. Technology is helping us bridge a lot of those gaps. But especially in the software world where salary is a lever, but at a certain point, there is a law of diminishing returns on salary in terms of incentivizing an employee. Once you get to a certain salary, raising it another \$10,000 a year is like, "Nice! But does that changed my quality of life as much as I get to live or I want to live in the world?" There's other leverage you can pull and quality of life is a huge one that I think people are starting to say, "Yeah. Actually, if quality of life is more important to me than a title or a salary."

[00:33:21] JM: Okay. Thinking further about radical models of knowledge work. This is why I kind of like this conversation, is because software development and knowledge work are in their infancy relative to a blacksmith or assembly line worker. I mean, there was a time when assembly line work was totally new. It was totally novel. Now we think about an assembly line, "Oh, it's just a way of doing work."

[00:33:43] JS: It wasn't that long ago either.

[00:33:45] JM: It wasn't that long ago. You have this brand-new that, in the scheme of things, thing called knowledge work, and this even brander new way of doing knowledge work where you're working from home. Other radical models that should be explored more? There was one topic we did a show on called Programmer Anarchy, which is the perspective you can just hire a bunch of software engineers. You put some OKRs and some KPIs in place and basically the developers figure out everything. The developers do marketing and design and sales and everything else and maybe you attach bounties to things. Like, "Oh, if you do a good enough job, then maybe you get rewarded with the bounty or something like that." You could gamify it.

What are the extreme models that we need to try more of within the software ecosystem?

[00:34:36] **AS:** It stumped me.

[00:34:37] **JS:** I'm stuck on this programmers do everything model.

[00:34:40] **JM:** You guys derive satisfaction from doing a little bit of everything, right? You're entrepreneurs.

[00:34:45] **JS:** Yes.

[00:34:45] **JM:** I've met many people who feel the same way and it seems like once people go in the direction of becoming an entrepreneur, it's rare that they revert. Why would you? People enjoy the cross-training. They like a little bit of everything.

[00:34:59] **AS:** It goes to, again, success. If you dabbled in entrepreneurship and failed and you're not resilient to failure, you may not see that as feedback as how to iterate towards a future entrepreneurship opportunity. That may deter you and you might retract and you might regress back into a non-entrepreneurial mindset.

But I think once you've sort of hit a little bit of success and you have some resiliency, then you're going to be the kind of person that can push forward and try these more experimental ways. The way we've designed our businesses is lifestyle. I just had a kid in December. Jerod's carried a ton of the weight in December. He hasn't complained once. Thanks God, right? But the point is

—

[00:35:43] **AS:** But the point is, is that we – The same thing when Jerod had his recent daughter. What was that? About a year ago, Jarod? Just a little past year ago? From the same timeframe, right? Was it January, February?

[00:35:54] **JS:** No. It was December 7th. Yours was December 10th. Three days. A year and three days.

[00:35:58] **AS:** Okay. Wow! Very close together. Same time frames. Same. I carried the weight and I wasn't angry with Jerod in any regards because we were in a family-oriented business.

We've designed the outlook of our business and even the choices we make around our business. How does this not only impacts our paychecks and profitability and revenue stream? But more importantly how does it change our family lifestyle. How does it impact our families?

I would say not just his and mine, but collectively even. We have some particular direction when we're doing it and that extends also to everyone that works with us that we act a lot like a family. We treat people well. We take care of things for them. We don't expect them to deliver more value to us than we give back to them. We went to reciprocate when they're in hard times or they're in a burnout stage or anytime where they need to take some time away. We're not like, "Well, you're committed to this. You got to come to this show and do these different things." We just say, "Okay. No problem. Do what you got to do. How can we help? How can we be there for you to support you? What can we do to help you through this time?" Rather than say, "Get back into a position to produce a show or ship a thing or create some software or deliver new interface." It's very flexible. That's a kind of business I want to run and work for.

[00:37:13] JM: I feel roughly the same way, at least in terms of the freedom that you have. I mean, if you're working for yourself, working from home, lifestyle business kind of thing, you have freedom. You can do with that freedom what you want. If you want to amp up your family life or focus on family life, you can do that. If you want to amp up your business or you want to go into some hobby, you want to travel, you can do all those different things.

What's funny is in our particular seats, most or at least many of the people that we interview are in a totally different situation. They are in a working for these companies to working for a big company, a small company. There's a particular kind of satisfaction that you get from working on those big teams from shipping these hugely influential products that you basically can't ship without being on these closely integrated teams.

I wonder if we're missing something. What are we missing? Maybe there's some kind of validation you get from working on these big software teams, shipping these huge products that we don't get a taste of anymore, or never did.

[00:38:23] AS: I definitely thought about that. First of all I should say that our industry is so filled opportunity that we consider and think like what kind of a life do I want to have? Because that's

really what you're asking yourself. In my career, do I want to maximize profit? Do I want to maximize freedom? Do I want to maximize impact or some sort of balance of those three or maybe there's other attributes of my life that I am going to design?

It's amazing, the opportunity in our industry, where we can have these kind of conversations, because business is really hard and staying in business is really hard and yet there's so much opportunity inside of software and technology that many of us can excel and stay in business or sell our business.

I've optimized my life for freedom from very early on. That's what I'm after. I got it early on. I was very fortunate to have the freedom early on, and now I regard as precious. To have that freedom, I have to trade other things. All things are tradeoffs. What I don't get is I don't get impact, because – Like you said, I talk to people on our shows and they're doing these things. They're creating projects that are affecting so many people. My poster child is Daniel Stenberg of cURL. Maybe he has both. Maybe he's doing better than us. He's got – He might have freedom and impact. That's one thing about open source, is you can have impact as a single contributor.

The people who work on the notes app for Apple's iOS or for Android, right? Think about how many people use notes.app on iOS in the world. Millions. Maybe – Not a billion. That I think of a billion users. If everybody uses the default notes app, your software is helping millions and hundreds of millions of people. That's impact that I'm not going to have, because that's a tradeoff that I've taken. I think, yeah, we are missing out on things, but we're fortunate because we get to choose those things.

[00:40:23] JM: Last year you guys did some shows with a few classic software voices, Bob Martin, Dave Thomas, Andy Hunt. The subjects that these individuals have explored, how durable are they? These guys are writing books that are classically referred to as part of the canon of software development. How canonical is their material? Are they sacred cows that are durable or are they sacred cows that perhaps are outdated and maybe should be slaughtered?

[00:41:01] AS: My answer would probably lie in the fact that they're our most popular episodes of the entire year and most of them were produced and shipped in the latter half of the

year. I think post-July, pragmatic programs was – Let me just look at the timestamp real quick. My memory – Yeah. Right there at the middle mark, July 11th. [inaudible 00:41:20] Bob Martin. I think that was like November or October or some point at that. Even later. Yeah, October 31st. Halloween, basically. Those are two of our most popular episodes last year. We just based on our stats alone.

I would say they're desired, they're obviously popular. If they're canon, I think they're already canon in that regard, but I think they're timeless topics. You can always – I suppose in an agile world, I was talking about agile from a Bob Martin perspective and get and raise many waves. If you're Andy Hunt and the pragmatic programs, if you're those guys, then you can easily sort of revamp a 20-year addition of your collective knowledge and still see the same kind of waves, because it's such canon already. You're sort of revising it to fit the next decade and beyond potentially. They'd even said that the last 20 years, there have been so many changes in the way software is produced. Not a lot has changed, but also plenty also has changed. A middle ground there.

[00:42:18] JM: Bob Martin and pragmatic programmers, author pair respectively. How would you boil down each of those philosophical treatises? What is the thing to take away from each of those pieces of canonical work?

[00:42:34] AS: I'll make it easy. Just go listen to this episode. I mean, we also have a transcript. If you're a reader, just go ahead and read it. That's your take away. That is about an hour-ish investment per and draw your own conclusion about it.

I think back to agile base, is what I love about Uncle Bob is he shares the sort of 50-year veteran in software wisdom. One thing that occurred to me during that show was how often do you get a chance to sit down with somebody in our industry and have a 50-year-old experience-based conversation of wisdom? It's not often.

He even said with the doubling and the effects of the community growth, the software developer community growth, people that had been in software at his age bracket. I think he's 67? Correct me Jerod? There's not a lot of 67-years-old out there that started their career in software. There's a very small pool of Uncle Bobs out there. My tried-and-true opinion has been to listen

to people who've been there, done that for many, many years. I think when you hear him speak about the way software is done – And maybe in some cases he's wrong, but in a lot of cases he's right.

Worst case, you're getting some pretty interesting wisdom along with some antidotes and some data points, because he shares stories from other people. It's not just simply by Uncle Bob. He's invited others to write chapters as well and added some flavor to it. But it's not simply just Uncle Bob. It's a little bit more than just him and you have that wisdom that I think is not found often in our industry. By wisdom, I mean, not that kind of smart, but that kind of depth. That level of 50-year plus experience.

[00:44:05] JM: Over the next 10 years, are we going to see software become a trendy art form? Is it going to become like music or video where it is as trendy as pop music?

[00:44:20] AS: Isn't it already? Look at Glitch now. No. That's not. Look at Tik-Tok versus Glitch. I mean, massive size differential there. I mean, music is global. Not global impact in terms of accessibility, but in terms of people who are involved in it. Billions of people are involved versus millions. Yes.

[00:44:39] JM: Yeah. The – I don't know. Perhaps the glamorization or the intersection with pop culture and influential people and the aspirational side of it.

[00:44:52] JS: I hope so. I think that over time, these worlds blend. We already see the blend of software and music. Jeff, you know that very well as a musician and somebody who's into software. There is a lot of overlap in video games. There's a lot of overlap. Lots of the different pop cultural areas, movies, criticism, video, etc.

I think that programming has to become much more accessible to a general audience for it to be wielded by a general audience. I think with every layer of abstraction, we kind of move towards that end. I heard you speaking recently with Kevin Ball about the No Code Movement. Wan we provide programming interface to people who don't have to study and put years into it in order to produce something beautiful and amazing? I think kind of sort of, maybe we're going to get there. Kind of like that's where it is and we're trying at that.

I definitely think there's – You see a mirror between – I mean even in way that recruiters talk, like rock star developer, like ninja. There is already a glamorization to a certain degree of people who can sling code. I think as more and more people can do that and from their basement in Idaho, write some code that improves the lives of somebody who's over in Japan or like people or all around the world. That amount leverage is currently available in music. You can write a song in Russia and it can speak to me over here in Nebraska. I think that that, they're accessible to the tooling, I think that that will probably happen. I think that will be glamorous.

[SPONSOR MESSAGE]

[00:46:51] JM: Today's show is brought to you by Heroku, which has been my most frequently used cloud provider since I started as a software engineer. Heroku allows me to build and deploy my apps quickly without friction. Heroku's focus has always been on the developer experience, and working with data on the platform brings that same great experience. Heroku knows that you need fast access to data and insights so you can bring the most compelling and relevant apps to market.

Heroku's fully managed Postgres, Redis and Kafka data services help you get started faster and be more productive. Whether you're working with Postgres, or Apache Kafka, or Redis, and that means you can focus on building data-driven apps, not data infrastructure.

Visit softwareengineeringdaily.com/herokudata to learn about Heroku's managed data services. We build our own site, softwaredaily.com on Heroku, and as we scale, we will eventually need access to data services. I'm looking forward to taking advantage of Heroku's managed data services because I'm confident that they will be as easy to use as Heroku's core deployment and application management systems.

Visit softwareengineeringdaily.com/herokudata to find out more, and thanks to Heroku for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:48:23] JM: I need to talk to you guys some about podcasting, because that is something we share in terms of our day-to-day activities. One thing that you guys have that I have never really experienced is the cohost. Describe the dynamics of a cohost of a podcast. How are you an effective cohost and what does the listener want from the foil of a cohost?

[00:48:50] AS: I think it has to be enjoyable. I think in a lot of ways it's a partnership at a very basic level. It may not be a business partnership at first, but it would deftly be a partnership and it has to be reciprocal, meaning that person has to care about me and have to care about them at some point to some degree. Maybe that's not always the case, because there may be like complete opposites and that's what makes the art form, the delivery of whatever you're doing, podcasting on could be entertainment-wise.

I think there should be some sort of chemistry, whether it's anti-chemistry or chemistry, meaning that getting along – Or not getting along. If it's the entertainment value of the podcast, if it's what it's trying to deliver, than the cohost needs to sort of match that.

I think it's also important to have a couple size to a coin. What coin do you find that doesn't have two sides? It's an interesting as well where Jerod may represent a different side, of in particular, the change log. I represent another side and we complement each other in that regard. But I think respect, and I think listeners want to have that engagement between one another too. It's important.

[00:49:53] JS: Yeah. I think in terms of what it provides is it provides perspectives. Two different perspectives can come up with I think ultimately more fascinating conversations with a guest than if I was just by myself, my own perspective, because Adam will ask things that I never would've thought of, because I'm just not even bringing that perspective whatsoever. Sometimes that will change the course of an interview or a conversation that we hadn't planned, but now it went in this whole new direction and this direction is actually kind of fascinating and an amazing. Sometimes I'll do that as well.

I think that's why it gives us perspective. It's also nice to feel conversational like more than just back-and-forth, but what it requires is trust and respect and a certain amount of humility to not be like, "This is my interview. I'll ask the questions around here." We have to think about each

other constantly. Sometimes I'll be like, "Dang! I'm talking too much. I need to ask Adam if he has anything he would like to get –" From just filling the void, I'll add a back channel like, "Adam, hop in here," because I care about his voice being heard and his perspective being shown. I don't want to hug the ball. It requires trust and respect and it provides multiple perspectives that you may miss.

[00:51:11] AS: I think a lot of what happens between Jerod and I too is very automated these days. It's just second nature for us. Even having to like retrospective and think about those things in particular is hard, because we've sort of automated a lot of what we do already. You could probably even see where there's – After you ask a question, Jeff, there's a slight pause between – It may get edited out in your version of it. Who knows? But there's some level of a pause, or in this case, now we have the video aspect where we're using Zoom. So it's a lot easier to have an empathy position because I can see Jerod's face or his demeanor. I may even see him puzzling over something and I can see visually the pause along with the audible pause.

A lot of that I think between Jerod and I specifically on the Changelog is just automated these days, because we've just barked together for so long. We almost know –Jerod will know when somebody is talking about something with space, physics or granularity of icebergs that I might go into the very, very small or the very, very large, because I think about physics a lot. That's just an interesting thing to me. I find some interest in that. He knows that I'm probably going to jump in there, and that's just the relationship.

I say chemistry, because that's a natural thing you have in a relationship, is this idea of chemistry, this commingling. That's interesting, I think. The humility, Jerod, I think is an important aspect to consider too, because it does take humility to not make it about you. You definitely put your ego aside when it's a cohost position and not feel like you're hogging or not offering.

[00:52:43] JS: Have you ever considered a cohost, Jeff?

[00:52:47] JM: This gets into a discussion of like –

[00:52:49] JS: I know you're a huge fan of the one-on-ones.

[00:52:51] JM: I do like the one-on-ones, but this gets into a discussion of like how do you scale a media business? How do you scale a podcast business particularly? It's come to the decision of like what do you want to do with that scale?

I mean, do you want to like just sort of grow your business in terms of like doing the same thing but hopefully getting better at it and maybe charging higher ad rates, maybe improving the audio quality, or like what you guys have done. Do you want to grow horizontally? Do you want to have additional shows? Do you want to have cross promotion between those shows? Economies of scale between those different shows? Do you want to go to video? Do you want a broad media organization like NPR? Where do you want to scale? What do you want to scale? Do you want to go to video? Do you want to go to conferences?

I think if you bring on a cohost, you're making a certain declaration about what you want to change about your show and how you want to scale. The current format works, it works for me. It's something I like to do. If I were to bring on a cohost, it would be to a point of changing the dynamics of the show.

I think, frankly speaking, part of it would be because I have such kind of high-demands for what I would want out of a cohost, I would need to compensate them. I would have the expectation that like you're going to bring to the table the same level of discipline than I am. That's a high bar.

[00:54:22] AS: Jeff, I think the answer to that idea is it's changed over the years. If you'd asked me let's say in 2016, 2015, my answer would be different than it would've been 2017, 2018. I think we've just kind of iterated and layered on and we've tried things. We thought that one show would be great and that'll be fine. Then we realized we wanted to kind of go out and have more voice. I've always looked at you as well as a consistent five shows a week, one a day platform and have always been like, "Wow! That's really hard." We've kind of gone there by diversifying our shows versus adding more episodes to one podcast. I'd say my answers changed over the years. Now it's a little different.

[00:55:07] JM: What about you two? I mean, you guys could have gone to video. We've talked a little bit about conferences. Both of our businesses have focused pretty particularly on

podcasting, going deep rather than broad. But why not do these other things, or is there any intention to eventually do the other things?

[00:55:25] AS: Something I'd actually – I don't know if it's a thing, if it's a written thing. Maybe I should write about it if it's not, but I've had this kind of idea of profit to effort ratio. We've tried other things like video and have done video things, but the effort was very large and the prophet was mediocre or not as big.

As any business might want to begin and to get some foundation beneath it, which is revenue, and I would even say reliable revenue. We've decided to focus what I think on is the profit to effort ratio. We focused on things that are higher profit that require less effort. Because we have limited bandwidth, we're just two people. I just mentioned earlier to about being very family focused and that's a very important aspect of our business.

Jerod and I are working – I don't know how many hours Jerod even works. That's 50 or 60. I don't think it is. He probably works just works 40, but I don't want to work 50, 60 hours a week. There are some weeks I might, because there's some ebb and flow in our business. Generally, we're trying to just be 9 to 5, 8 to 6, whatever the number is. My Northstar in my life and as well in business is work 8, play 8, sleep 8. If you do that, quality life will prosper. You'll enjoy your family, you'll enjoy the work you have, you'll enjoy the playtime you get to have, and all work and no play makes Jack a dull boy. Wisdom from Kubrick.

[00:56:53] JS: I can't find the quote. I've been Googling frantically over here. But there is a sentiment which I think I may have shared with you at OpenCore Summit, Jeff, that somebody said somewhere, and I've been parroting, which is that in – I think it was in 2019 it was said, but now in 2020, that the ideal size of a media organization limits to one. It actually reduces down to a single voice or a single face or a single person versus the old style, which was like how big can it get? I think those companies are struggling.

[00:57:24] JM: Except not the New York Times, not the New York Times. Not Washington Post.

[00:57:27] JS: Right, but there's only one New York Times. There are hundreds, if not thousands of other media companies that are either downsizing or failing or being merged, etc.

Whereas today with the leverage of the Internet and the easibility of production and really the platforms is that we have, your mark has Brown Lee's on YouTube or your Joe Rogan's. That model is great in terms of scale. You can scale your business without really scaling your business in terms of employee count or payroll.

We've thought about hiring, like expanding, like what do we want to be and who do you want to be when you grow up? In a large way we're kind of regarding the things that we have, which is our freedom, which is this way of doing business that we like. I think Adam's ideas around higher profit for lower effort allows us to stay in that wheelhouse and not have to expand, because we have very labor- intensive things to do. That's working very well.

That being said, you always want to kind of do more. You want to be more. You want to get in the video. I mean, I have ideas for I think great developer-focus content in video that I don't think anybody's doing yet, but we have bandwidth constraints. So then you have to start making hard decisions, and mostly we just deferred some of those things in order to build out our horizontal portfolio of shows. We aren't the only voices on those shows, like Adam said. We have contractors and relationships with people who hop in and out, and that's fun. It keeps it fun. It's not just the two of us in our own offices all day. We have other voices. We have other people.

[00:59:08] AS: We have coworkers.

[00:59:10] JS: It's interesting. They're not employees.

[00:59:13] JM: All right, last question. Predictions from each of you for the next 10 years of Changelog.

[00:59:17] AS: Listen to the episode we did with Quincy. I'm just saying – I'm just getting –

[00:59:22] JS: [inaudible 00:59:22] on this. Just go listen to our show.

[00:59:23] JM: I defer to podcast, started recorded to some degree.

[00:59:26] **JS:** It hasn't been answered.

[00:59:28] **AS:** Well, you should listen to that show anyways. If you're listening to this and you're this far into this show and you're this curious about what Jerod and I are doing or either of us in particular, it would be a service to listen to some of our industry. Probably understand a bit more about our roots and DNA and why we have the opinions we have, and if those opinions resonate with you, then reach out and say hello or whatever.

But I would encourage you to listen to that episode. You can find it at changelog.com/10. You can also celebrate with us, this decade we've spent podcasting, creating conversations, news and community. It's been a big deal for us to be a part of that too to sort of cultivate this idea of community and this idea of open source software and this importance of what we do. Enriching the lives of the developers as we say. That's a big deal for us. We're hackers to the heart. Jerod, I'm just using a lot of our taglines, man.

[01:00:13] **JS:** When Adam runs out of things to say, he just starts using our tag lines.

[01:00:16] **AS:** I do have more to say tough on that front. I think that for the next 10 years of Changelog, I think you probably see this do a lot more. I think what we've done is figured out some of our sweet spots and to see us chisel out further, which is to produce shows that are community-focused. Producers shows that aren't Jerod and I or simply just Jerod and I that produce also shows that he or I specifically have vested interest in. If we bolted on a show that we had no interest in, it wouldn't do our network or what we do any service, because we can't be personally invested in them.

I think a personal – We've learned that a personal investment of either he or I is kind of crucial to its success. If I don't care about it and Jerod doesn't care about it, if falls by the wayside and nobody cares and we're the only ones around here picking up the ball first. We hand it off. I'd say that's probably how we'll build our business. In terms of other things, we'll probably try and experiment, iterate and kind of go back to that profit to ratio and say, "Does it have a degree of profit for its effort?" If the answer is no, not right now, then we'll just put it back on the shelf and come back to it later.

[01:01:23] JM: Jerod, any predictions of your own?

[01:01:26] JS: I've never been much of a planner. I can't see three months ahead of me, let alone 5 or 10 years down the road. There I am. I'm agile. There's my sacred cow. I honestly don't know what media is going to look like in 10 years, so let alone Changelog media. It's kind of really hard to predict without looking like a fool with my pants on the ground.

[01:01:44] AS: I have some hopes. I sure hope that podcasting continues to be a form factor for which people gather knowledge. I hope that it, for sure, remains a key piece of a software developer's toolkit to reach into and to be entertained, to be educated, to be a part of a community to have a voice and to listen to other voices, part of be like or learn from. I hope those two things remain true. If they don't, then our business dramatically changes. We get different problems.

[01:02:14] JM: Join the club. All right, guys –

[01:02:19] AS: Can I ask you a question then, Jeff?

[01:02:21] JM: Yeah, shoot.

[01:02:22] AS: You've known us for a number of years. You asked us these kinds of questions and I'm curious as an outsider and observer to Jerod and I and Changelog media, what do you think we should do in the next 10 years or try?

[01:02:35] JM: Well, frankly speaking, I think it's the same story for me. It's change or die. I don't think that podcasting is going to be around in 10 years, at least in its current form. I don't know what that change looks like. I just did a second interview with Tim O'Reilly, because I think if Tim O'Reilly is basically the gold standard for what our business is, right? I mean, nobody has been more successful in the world software media than Tim O'Reilly.

I think it's like the positive trends that we have to take advantage of are just that the market is growing and growing and growing. You have an opportunity, you have a really strong brand. Changelog is a really strong brand. It's a question of like what do you want to do with that

brand? I don't know exactly what that looks like, but I think you want to position yourself to basically have some competency in these other categories, whether it's YouTube, or whatever else you want to get into.

You guys certainly do Twitter better than I do. I'm terrible at Twitter. I don't have any – I don't have any twitter strategy basically. My strategy is to not be on Twitter, basically. I mean –

[01:03:42] **AS:** Tell him your strategy, Jerod. This is Jerod's credit. Not mine.

[01:03:46] **JM:** Memes. One word.

[01:03:47] **JS:** Yeah. That's it. I mean, honestly, I'm not even a huge fan of memes. It's just like what works on Twitter, it's memes, and I can scratch together a meme.

[01:03:56] **AS:** He's very good at that. He can look at a video and derive some way, shape or form that it relates back to the software developer lifestyle and/or processes and/or ways. It's really interesting. I'm personally entertained by Jerod's memes on our Twitter account.

[01:04:13] **JS:** Well, thank you. I have one fan out there.

[01:04:16] **AS:** And they get lots of retweets. Wow! I mean, there's been ones that's like 10,000 retweets and a million views or something like that. It's like crazy that these things get that much visibility. It's well with that network effect does to one single tweet and it's from a dude on a computer in Omaha, Nebraska.

[01:04:34] **JS:** Twitter is weird.

[01:04:35] **JM:** Yeah. But what I will say is I think it's going to get harder. I mean, it'll get easier. If gets easier, that's great. Like is great.

[01:04:41] **JS:** Well, what did Tim O'Reilly say that led you to these? Because you said you talked to him, but I don't think –

[01:04:45] JM: Oh! Right. The interesting Tim O'Reilly thing is O'Reilly is a story of three business lines, each one subsequently bigger than the next. The first one being the print media business. Second one being the conference business, and the third one being the e-learning business and the idea of laddering over basically 40 years from a completely bootstrapped company. Over 40 years going from a documentation company, to a book publishing company, to a conference company, to an e-learning company led by this guy who just has tremendous hustle.

That to me is something – I don't know what the lesson is, but there's some lesson there. I see other players – It's amazing how differentiated O'Reilly is from the other people who are in this space, and maybe it's because he started earlier. Maybe it's because –I don't have the exact understanding of it. I wish there was a book I could read on the history of O'Reilly. Yeah, I don't know. Only the paranoid survive. So I think we should be paranoid. That's my perspective. I want to lose my business viability.

[01:05:51] AS: Can I share my biggest fear?

[01:05:53] JM: Sure.

[01:05:54] JS: Spiders.

[01:05:55] AS: Yeah, spiders. Arachnophobia. Well, as you speak about Tim and the kind of businesses he built, it reminds the businesses that I don't think I want run. Not that I want to be a large-scale software media company. I don't personally aspire to run a business that has many, many conferences and that kind of –

[01:06:13] JM: But it's better than having than no business is what I'm saying.

[01:06:15] JS: Your biggest fear is that you might be a too successful business?

[01:06:19] AS: This is my biggest fear then, at least, so far. I guess the easiest way to describe it is just that, to be "successful", whatever that means for this business that it would require us to create a company that I do want to run. That's my biggest fear.

[01:06:34] **JS:** Like get bigger or disappear.

[01:06:37] **AS:** Right. Exactly. Either going to obscurity because you're not willing to make the necessary changes, that's probably my biggest fear, because I like the kind of business we run. I hope we can run it for another decade, if not more. I hope that, as I said, that the way that podcasts are received and the way that knowledge is shared especially with software developers in this ecosystem wherein I don't care if other ecosystems change. If software developers always care about what podcasts bring to their life and they always add value, then we'll have a home. But if the way things change media-wise force us to create a business to be "successful" is a business I don't want to run, then it would be unfortunate.

[01:07:18] **JM:** I'm with. I hope that doesn't happen. I mean, I'll be in the same boat. Look. Trust me. I'm just scared. I don't know. Maybe I'm too scared.

[01:07:27] **JS:** If podcasting fails, I think I'll just do what we were doing before we started podcasting. I'll just keep writing software. There're so many opportunities.

[01:07:36] **JM:** Here's the thing, is I think after you – This is why I think it's pretty awesome to do the cross-training involved in starting a business. Consulting, like leadership training, software management training. There's something there. There some kind of business there. There's something to fall back on. They're something to pivot to, especially if you were under the gun. There's something. Unless the economy has some macro-failing, in which case we're all in trouble. There's no way to prepare for that situation.

I feel like there's some fallback. Just the breadth of knowledge and the differentiated position that you get to by virtue of interviewing people for a weird podcasts, a very niche podcast for 10 years, you're differentiated in a way that people are going to pay you money for a very, very long time is my perspective. But I'm just paranoid. I'd never want to – I don't ever want to work for a big company again. Just that's the thing that scares me. You want to know my biggest fear?

[01:08:36] **AS:** I was going to say. You could just go back and get your job back at Amazon, right?

[01:08:40] **JS:** You should see his face listening. You should see Jeff's face.

[01:08:43] **JM:** It's not Amazon even. It's the interchangeability of the different companies that I worked for. Just the servility and the tedium and the –

[01:08:55] **AS:** The games, the politics, the insider stuff.

[01:08:58] **JM:** The commute.

[01:08:59] **AS:** Yeah, the commute. Well, that's if you had to go into the office.

[01:09:01] **JS:** What would you listen to on the way? You have to listen to Changelog, I guess.

[01:09:04] **JM:** Oh! I listen to plenty of Changelog. You guys! I was at inbox zero of Changelog, SERadio, This Week in Startups. This is why I love podcasts, is like it was like nonstop clockwork orange for my ears level podcasts inputs when I was just figuring out how the hell am I going to escape this tedium. Anyway, that was me.

[01:09:30] **JS:** Now you've escaped and there's no way you're going back.

[01:09:32] **JM:** I hope not. Hope not.

[01:09:35] **AS:** So then let's give a command then or a request or a desire to the listening audience of this. Keep listening! If you want to help us or help Jeff, the easiest way you can help is to go and tell two friends. Just to friends. Literally, open their phone. Literally, show them the podcast app, if they don't already know. Show them how to download it, because not anybody knows.

[01:09:54] **JM:** Right. That's true.

[01:09:54] **AS:** Show them your favorites. Share your favorite episode.

[01:09:57] **JM:** Here's the other thing. I don't even – Some listeners asked me to set up a Patreon channel. I set up a Patreon a long time ago, but I don't want people to promote it. Please do not donate to me on Patreon. Because I want to be in a position where I can say, "If the bottom drops out of the podcast market, maybe I could just say, "Hey, look. Please, 1000 true fans, I need you now."

[01:10:17] **AS:** Oh! That'd be that big moment. Yeah.

[01:10:19] **JM:** If the bottom drops out, that's when I ask for the charity.

[01:10:23] **AS:** There's your fallback plan right there. Charity.

[01:10:26] **JS:** I've entertained you for these many years. Please. Please buy me Ramen noodles.

[01:10:31] **JM:** That's right.

[01:10:31] **JS:** For a month. Oh, wait. A quarter.

[01:10:34] **JS:** I'm usually the pessimistic one, but amongst the three of us, I think Jeff's taken the cake here.

[01:10:38] **AS:** Yeah, he really is. He's scaring me, honestly. I'm never talking to you again, Jeff. This is the last time.

[01:10:43] **JM:** I probably told you guys this, but like it's because I used to play poker, and the bottom dropped out of the poker market. Poker market collapsed completely, and it's just not a fun experience. I never want to go through that again.

[01:10:54] **AS:** Having said that though, were there any – In retrospect, were there any telltale signs that it was going to – That you did not see then that you see now that you can sort of identify as potential failure points for current scenario?

[01:11:08] **JM:** Sure. I'll tell you. It was a very relaxing job. I get to make my own hours. Sound familiar? Sound familiar? It was a booming industry.

[01:11:19] **AS:** Then what happened?

[01:11:19] **JM:** Felt like it was getting more and more popular.

[01:11:22] **AS:** The World Series of Poker.

[01:11:24] **JM:** World Series of Poker.

[01:11:26] **AS:** It was in the limelight for a while there. It was a regulation though. Wasn't it?

[01:11:29] **JM:** It was regulation. That's basically what killed it. I mean, poker provides much less value than software. I honestly think – Part of me knows that I'm way too paranoid.

[01:11:38] **AS:** Who's going to regulate the podcast market?

[01:11:41] **JM:** Spotify? I don't know.

[01:11:43] **JS:** I would say aggregators and indexes. That's what regulates –

[01:11:47] **AS:** Yeah. I think if we had to have a fear point potentially, it would be if Spotify decided to not acquire, but to destroy sort of their – Part of their growth strategy is to bolster up podcasts and then chisel into each different genre, which basically gets easier, because you just go to the topics that are within iTunes and how that works. The way that podcasts are categorized.

[01:12:10] **JS:** Categories.

[01:12:11] **AS:** Yeah. I mean, it's pretty easy to sort of say, "Okay. Once I've conquered one level of the category, go to the next, and to the next." If they decide to do something like that and create original content, if we have the same kind of [inaudible 01:12:23] in the streaming wars in podcasting, which I hope – I guess there's probably some telltale signs there. I would desire to be acquired rather than destroyed.

Just saying, Spotify, if you're listening to this, acquire, not destroy. Actually, that's a good [inaudible 01:12:39] acquire, not destroyed. Then put a little face of Jeff down at the bottom saying, "I'm just paranoid."

[01:12:47] **JM:** That'd be cool.

[01:12:47] **JS:** Or yeah, I'll wear that shirt. I'll wear that shirt.

[01:12:48] **JM:** All right. Guys, thanks for coming back on the show. It's always a pleasure and we'll do another check-in before too long. We need to keep doing this – I don't know. The conversational aspect of it. I'm trying to learn from you guys, the conversational, rather than the robotic questionnaire that is the format of my podcasts for most episodes.

[01:13:07] **JS:** Hey, man. Your show is beloved. Every time I listen to it, I learn something. What you're doing is working as well. Different styles, nothing wrong with. But it's always fun chatting with you. We appreciate you having us on, and we'll definitely do it again.

[01:13:19] **JM:** Cool. All right.

[01:13:20] **AS:** This is fun, Jeff. Thanks.

[END OF INTERVIEW]

[01:13:31] **JM:** Apache Cassandra is an open source distributed database that was first created to meet the scalability and availability needs of Facebook, Amazon and Google. In previous episodes of Software Engineering Daily we have covered Cassandra's architecture and its

benefits, and we're happy to have Datastax, the largest contributed to the Cassandra project since day one as a sponsor of Software Engineering Daily.

Datastax provides Datastax enterprise, a powerful distribution of Cassandra created by the team that has contributed the most to Cassandra. Datastax enterprise enables teams to develop faster, scale further, achieve operational simplicity, ensure enterprise security and run mixed workloads that work with the latest graph, search and analytics technology all running across hybrid and multi-cloud infrastructure.

More than 400 companies including Cisco, Capital One, and eBay run Datastax to modernize their database infrastructure, improve scalability and security, and deliver on projects such as customer analytics, IoT and e-commerce. To learn more about Apache Cassandra and Datastax's enterprise, go to datastax.com/sedaily. That's Datastax with an X, D-A-T-A-S-T-A-X, @datastax.com/sedaily.

Thank you to Datastax for being a sponsor of Software Engineering Daily. It's a great honor to have Datastax as a sponsor, and you can go to datastax.com/sedaily to learn more.

[END]