EPISODE 356

[INTRODUCTION]

[0:00:00.5] JM: Software Engineering Daily has been around for almost two years. In this episode, Pranay Mohan and Erika Hokanson join me for a reflection on where we have been and where we are going. Pranay was the producer of Software Engineering Daily for the first year after which he left to work as an engineer at Snapchat. Erika joined the show nine months ago to work on operations and ad sales, expansion plans, editorialism.

The thesis of Software Engineering Daily has always been that serious, in-depth material about software provides value. Right now, we are a podcast about software engineering. We are planning expansion into a larger media company with video, a mobile app, desktop platform, more podcasts, more journalism. In the meantime, we need to focus on the quality of the podcast. We want to keep that high as we gradually expand into other mediums.

Pranay and Erika and I have crafted the vision for Software Engineering Daily and we just want to hear your input. You can always send me an email, jeff@softwareengineeringdaily.com. We want serious and inspiring and technical content to be more widespread and we know that you do to. In this episode, you're going to hear a little more about that vision and just how we think about what Software Engineering Daily.

It's already been 500 episodes and I'm even more excited about Software Engineering Daily than I was when I began. I'm looking forward episode 1000.

[SPONSOR MESSAGE]

[0:01:53.5] JM: Spring is a season of growth and change. Have you been thinking you'd be happier at a new job? If you're dreaming about a new job and have been waiting for the right time to make a move, go to hire.com/sedaily today. Hired makes finding work enjoyable. Hired uses an algorithmic job-matching tool in combination with a talent advocate who will walk you through the process of finding a better job.

Maybe you want more flexible hours, or more money, or remote work. Maybe you work at Zillow, or Squarespace, or Postmates, or some of the other top technology companies that are desperately looking for engineers on Hired. You and your skills are in high demand. You listen to a software engineering podcast in your spare time, so you're clearly passionate about technology.

Check out hired.com/sedaily to get a special offer for Software Engineering Daily listeners. A $600 signing bonus from Hired when you find that great job that gives you the respect and the salary that you deserve as a talented engineer. I love Hired because it puts you in charge.

Go to hired.com/sedaily, and thanks to Hired for being a continued long-running sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:03:23.6] JM: Welcome to episode 500 of Software Engineering Daily. Today, we are going behind the scenes for a short interview with Erica and Pranay who are the two people who have worked with me on Software Engineering Daily getting it off the ground, getting it up and running. We're going to just chat today.

Pranay is currently a software engineer at Snap and Erika still works with me on Software Engineering Daily. Guys, welcome to the show.

[0:03:53.3] EH: Thanks, Jeff.

[0:03:55.0] PM: Yeah, thanks. Thanks for having me, and big congrats to everybody here on episode 500. It's a big deal.

[0:04:00.5] JM: Certainly is. Pranay, you left Software Engineering Daily — Gosh! It must've been around episode 200 or 250 or somewhere around there, probably about halfway through. You left Software Engineering Daily because you wanted to become a software engineer yourself, and that motivation was hard for me to take at the time because it left me high and dry. As time has gone on, I have totally understood and respected your decision.

Describe the motivation to leave a software engineering journalism and become a software engineer full-time.

[0:04:42.1] PM: Yeah, I think this is a tough thing to kind of also wrap my head around because, understandably, it was a difficult decision for both of us and it always makes me feel bad when I have to recollect splitting up and kind of deciding to go about my own thing. I think a lot of it was similar to how Preethi Kasireddy felt when she left venture capital to become software engineer because she kind of expressed the sentiment of when she was a16z and seeing all these companies come in and all these talented founders and technical folks come in and pitch their ideas, there is a strong yearning within her to also embark on the journey, and I kind of got the same experience through being the producer of SEDaily.

When I edited episodes and listened to all these fantastic people talking about their technical challenges, the interesting projects they're working on, I knew that I also had that spark to want to dig in technically and work on stuff like they were working on. As much as I loved working on SEDaily, at the moment, the ability to be extremely technical and in the weeds with code was not available to us.

I think in order to fulfill that deep desire, I had to find another opportunity. That's the kernel of motivation where it came from. It's always difficult to transition and kind of leave what you're doing because it can, as you mentioned, deal with interpersonal difficulties. I think when you have a strong motivation to do something, you have to act on that source of truth.

[0:06:18.7] JM: Not to mention, I think you and I both intend to work together again in the future, and I think we left on terms that were basically we're both entrepreneurial or whatever word you want to use, and we have a desired to build things and change things and whether it's collaboration within the same company in the future or collaboration across companies, I have no doubt that we'll work together again in the future. It was bittersweet, but not terribly problematic, taking a long-term view of things.

[0:06:53.3] PM: Totally, and we were friends for years before and we'll continue to be friends for years. It's just more of when will we find our window again and work together. I strongly believe in that.

[0:07:03.8] JM: What's been the difference between reporting on engineering and actually being an engineer?

[0:07:10.5] PM: Probably the breadth of what I've been able to focus on has decreased, and that's been a challenge for me every day because when you delve deeper into the technical weeds, the details of what you're working on, you necessarily have to become more focused. You have to become an expert and deep dive into something. For me, that's been web development and JavaScript, so I've really, over the past six months to a year, I've really gotten to know the guts of react and redux, and these are concepts and ideas I was exposed to while working on SEDaily, but I was also supposed to microservices, Kubernetes, big data, Spark, a gamut of subjects, but as an engineer, you cannot focus on that many things at once and be working on a product at the same time with the specific technical stack.

That's been the biggest trade-off for me, but it's also rewarding in the sense that as you go deeper down the levels of abstraction, you learn more about the trade-offs, you learn more about what are the applications of the subject you're working on, and that's not necessarily accessible to you when you're just researching and learning about it on a journalist level.

[0:08:25.4] JM: Yeah. You and I talked about this a lot, that preparing for episodes and Software Engineering Daily has been largely a matter of knowing the right question to ask rather than having to have the right answer yourself. It's good to know that that's the focus when you're doing interviews about software engineering, that you know you need to ask the right questions but not being able to assimilate that information into a measurable skill, like being able to build a react-redux website even though you've — I can't do that even though I've done 10 or 20 shows on react. Something feels a little off about that exchange.

[0:09:13.2] PM: Right. I think it's also riding a bike. You and Erika are both software engineers, and given the desire or the choice to go into it, you can you can do it instantaneously. It might

take a little bit of time to wrap up in whatever specific field; frontend, backend, infrastructure, that you decide to go into, but is readily accessible given the desire.

I'd like to get y'all perspective on this as well, like having been software engineers and transitioned into being journalists, do you feel like the trade-offs are what I articulated, or do you notice something else about it?

[0:09:50.1] JM: Erika?

[0:09:50.7] EH: Pranay, there definitely are some trade-offs. I mean, the breadth versus depth analysis, I think about that a lot. I feel sort of like a generalist at this time, although I love doing reading and research, and just being involved in the industry, what everyone's building, all the different players. Lie you said, the opportunity is there to get back into coding and build some products.

We've talked about some side projects that we've had on the brain, Jeff and I, but over time, I think it'll just be following wherever the passion is. Right now, I think our passion at least is in bringing great technical content to the listeners of the podcast, expanding what the podcast is, and how we can bring great info to people and educate people, and also reach developers in the technical community outside of podcasting. Like anything, there are trade-offs, but I feel we're in a great place. It's really exciting time. Jeff, what do you think?

[0:10:53.6] JM: When Software Engineering Daily started to get a little bit or traction, I talked to Blumen who is as frequent listeners to the show. He's somebody who started Software Engineering Radio, and that's the show that Software Engineering Daily was originally based off of. I talked to him, and I was like, "Hey, I would love it if you joined me full-time and we did Software Engineering Daily together."

One thing he said was he wasn't sure that he will be able to report effectively as a software engineering reporter if he wasn't in the weeds day-to-day working on software products. I think there's some truth to that, especially when I started Software Engineering Daily, I did not have a lot of engineering experience, and you could hear that in the episodes.

Pranay, I remember, I think the first episode that you listened to of Software Engineering Daily or one of the first episodes was that episode with Matei Zaharia about Spark. As much as I have wanted to listen back to that episode, I cannot stomach going back to it because I know that before I did that show, I didn't know what Spark was, and after that show I didn't know what Sparks was.

Just like you said, this breadth versus depth thing, if you go too wide and you don't have enough software engineering background, which I really didn't, I just had some internships and some jobs that I hadn't done a great job at, that was my level of experience and that was really not enough to do a good job of reporting. I would say, since then, since I started Adforprize and started working on some other software engineering projects, getting back into the thick of things, even though I'm not writing code, working with engineers directly on software products has probably really —It's had a lot of synergy with my reporting on Software Engineering Daily.

[0:12:38.6] PM: I think, basically what you're describing is — I think this begs the question where is the foundation of software engineering such that you can cement that bedrock of knowledge and then move on to doing maybe tangential or corollary pursuits but be assured knowing that your foundation of knowledge is solid and that you can always draw upon it to help you with journalism or product management or design or whatever that secondary functions is.

I think that my opinion is that that goalpost is moving farther back, like it becomes easier to cement that foundational of knowledge, and a lot of people would disagree with me. A lot of people believe that the technical foundations of software engineering is in computer silence, you have to get that four-year degree, and we've talked about this a lot.

In practicality, the tools are becoming simpler, the languages are becoming more dynamic, expressive and high level. In all intents, it's actually becoming easier, I think, to cement that knowledge of being a software engineer and moving on to doing something else. I think it has to be in order for our society to be able to employ more people in this field.

As software eats more of the world, the ability to create software must get easier in order to bring more people into the fold. Otherwise, we're going to have this class of elite software wizards and then luddites and people who just interact with software but don't have an

understanding of how it works. That sounds to me like a less preferable world than the one where everybody's involved in making software.

[0:14:18.1] JM: Completely agree, and you probably see this firsthand at Snap because it's a design company as much as an engineering company, and I bet that the designers the you interact with have such a technical proficiency and they have a sufficient technical proficiency to know how to design. It probably almost seems like engineering, the type of work that they do. Part and parcel with engineering getting easier, it seems like other roles are getting more technical. How do you see the product design, the product engineering process changing from your vantage point at perhaps the most cutting edge software, well, at least consumer software product in the world?

[0:15:08.7] PM: Right. I can't speak exactly to what Snap does, but I will definitely agree with you that a designer that even has a remote understanding of react and how components are designed can really assist that process of building an interface.

I think, also, react has helped a lot with this because the concept of a component and being able to break out an app into these modular pieces that can also be designed in the same workflow really helps. If we're on the same page in terms of the component and I can talk about how the component loads and what the different interactions with the component are, then my workflow with that designer is a lot easier than if the only speak on terms of user interface design and then I have to translate that into even the most basic blueprint of how my app structure is going to be.

At any point when I'm working with the designer or product manager who has an exposure to the tools I'm working with, our language becomes in sync. I think that's the biggest thing, the language of communication needs to be a level of technicality that everybody agrees upon. I'm okay as a software engineer with making the language less technical. The designer does not necessarily need to know about react lifecycle or going further JavaScript garbage collection, but they should at least know enough to say that, "Oh, hey. I know that you're designing this nested component and I can think about how to make these components responsive when I'm designing it.

[SPONSOR MESSAGE]

[0:16:50.5] JM: Your application sits on layers of dynamic infrastructure and supporting services. Datadog brings you visibility into every part of your infrastructure, plus, APM for monitoring your application's performance. Dashboarding, collaboration tools, and alerts let you develop your own workflow for observability and incident response. Datadog integrates seamlessly with all of your apps and systems; from Slack, to Amazon web services, so you can get visibility in minutes.

Go to softwareengineeringdaily.com/datadog to get started with Datadog and get a free t-shirt. With observability, distributed tracing, and customizable visualizations, Datadog is loved and trusted by thousands of enterprises including Salesforce, PagerDuty, and Zendesk. If you haven't tried Datadog at your company or on your side project, go to softwareengineeringdaily.com/datadog to support Software Engineering Daily and get a free t-shirt.

Our deepest thanks to Datadog for being a new sponsor of Software Engineering Daily, it is only with the help of sponsors like you that this show is successful. Thanks again.

[INTERVIEW CONTINUED]

[0:18:14.5] JM: Erika, you joined Software Engineering Daily full-time about five months ago. That was after you worked with me part-time, specifically on the newsletter and some of the WordPress stuff, and your background prior to learning to code was English editorialism publishing media production. Pranay and I are now talking about how the design and the engineering roles are becoming the relationship between the two are changing.

I think this is true also in editorialism, in journalism, where you see the organizations that have an engineering vent succeeding more, companies like BuzzFeed, and then the New York Times, Washington Post, kind of working towards becoming more technical in their proficiency. What's your point of view on this having spent a lot of your time in publishing and editorialism and so on before gravitating towards this more technical world?

[0:19:26.1] EH: Working on the production of any kind of content has a standard workflow, but in communications for any industry, it's important to have a high level of knowledge and interest in that subject matter to present it in the best way possible and know the audience. So the more we immerse ourselves in the culture of programming and technical research, and keep at least one foot in the pool as developers, the better we will be at reporting on the tech industry and presenting accurate information to our audience. The show's slogan that maybe hasn't been showcased much is "The world through the lens of software" and that's the view that we aim to publish the episodes from and for.

For a few years before I came on board at Software Engineering Daily, I was learning to code and was listening to a lot of tech podcasts, so I was an early listener and huge fan of your podcast, Jeff, before we started working together. I was so thankful for the technical depth and consistency you provided with this show. Combining my prior experience in publishing with my technical interests has been a wonderful way to both give back to our tech community and stay current in a quickly-evolving world and I do think that those combinations are important in the media.

And Jeff I also want to say, thank you for this opportunity, working on this show with you is an incredible honor, -- I admire your talents as an innovator, researcher, and interviewer, and thank you to the fascinating guests who come on the show, as well as to the listeners and sponsors — this show is for you — and we welcome your input so please reach out. As Jeff has said, quality content is our priority. Thank you all for taking us to episode 500.

[0:20:19.2] JM: Describe some of the weirder nuanced aspects of working full-time on a podcast.

[0:20:26.2] EH: Well, one great thing is just all the amazing letters that you get from the fans of the show who are learning so much every day, and that really keeps us going, I think. As far as the weird stuff, we get a lot of interesting requests from companies and individuals, different shows they want to hear about, and sometimes things don't quite fit into the scope of Software Engineering Daily, but it's great to have so much interaction with the listeners and fans and bring related content as a result.

I guess there are a lot of quirks just where we — Our jobs are so multifaceted, I mean we wear so many different hats and making this happen on a daily basis, and I think just the key has been adaptability and an open-mindedness to those weird things when they come up.

[0:21:18.1] JM: Pranay, what did you find strange or contrary to your other work environments when you were working full-time on the podcast?

[0:21:27.9] PM: For me, the weirdest thing to me still is how approachable people are. When we were first starting out SEDaily, I was totally convinced that most software engineers, most leaders in their whatever specific expertise, whatever field they're working on, would not want to come on an obscure podcast for an hour and talk about what they're working on, but people really wanted to come on the show. They were excited about sharing what they were working on and having a conversation with you where you would surprise with interesting questions and they would be challenged on the spot and they were encouraged to think about what they were working on and think about the complexities of what they we're working on and then deliver it and give answers to a general audience who needed to be able to understand really complex technical subject.

That kind of insight into human psychology, knowing that people love to talk about what they're working on and people love getting challenging questions when the host is interested and genuinely wants to know more about what they're brought on to talk about, that was really interesting to me. I thought people were always more closed off, but that kind of reaffirmed my faith in people, in everyday people, because I think the fact that people become so passionate and interested in what they're working on is a sign that we are in a healthy world. We're in a world that is making technological progress, in a world that is trying to inspire more people to become technical and share their knowledge. I think that was the biggest take away from me, and in my career and professional life, I think I'm more encouraged now to reach out to strangers and subject matter experts and get their opinion on something because I'm not as afraid that they will be walled off.

Beyond that, the really bizarre things is the podcasting industry is super bizarre**.** I think you and I have found that out, just like the basic things that you would expect like having podcast analytics

and just having a level of sophistication I think is still not yet there because there still has this same underground fight club type of feel. That's what makes it so interesting, people — My experience has been when I share with people that I used to work on a podcast, used to be a producer, their eyes light up and they're like, "Whoa! What did you work on? That's awesome. I listen to podcast all the time." That conversation would never have come up unless I broke the ice by saying that.

I think there are a lot of people who consume podcast, but it's just for whatever reason, they don't talk about it in the mainstream or it's just thought to be like an esoteric thing, like reading a weird book by yourself. I'm still waiting for that shift in the podcasting industry where, if you can even call it that, it becomes an industry where people are more open about it and build tools that make the process more sophisticated. I think we're getting there especially because we are using this tool called Zencastr right now. Am I allowed to talk about that?

[0:24:31.6] JM: Oh, yeah. Please. We did a show on it.

[0:24:33.7] PM: Oh, cool. We're using a tool called Zencastr right now which when we started off, we were using Skype an Screen Recorder. The fact that this even exists and there are different startups coming out, I know there's a new one in YC batch called Breaker that's trying to make the best iOS app for podcasting. There's no one called Anchor that is trying to do podcast tidbits, like you send clips of sound to your friends and it's kind of like these micro podcasts.

I think we're kind of getting the bubbles to the surface of people exploring podcasting in different ways, and I really hope it leads to this explosion of really cool shows and a better listing flow for me so I don't have to go through iTunes all the time and manually download everything.

[0:25:20.3] JM: Regardless of the medium, whether it's audio content or written content or video content. Pranay, when you and I were talking over dinner or on Skype or whenever we would have all these endless calls about what Software Engineering Daily was and what it should become, we saw opportunities in the software engineering media landscape, or just the media landscape. What's the diff between what both of you want to see in media versus where we are today?

The media landscape overall, there's obviously so much turmoil in media, or at least there's — Maybe there's always been turmoil in media, but it's certainly gotten highlighted.

[0:26:07.3] PM: Fake news.

[0:26:09.1] JM: Fake news or just like the president attacking the media or whatever that means, whatever the media means, mainstream media.

Erika, why don't you go first? What do you think about the differences between where you would like to see media, the public discourse, and where we are today?

[0:26:27.2] EH: That's a big question in the world of news. There are so many news outlets available. I guess people have so many options today. There are a number of different news; TV channels, newspapers, apps, blogs, then you have all your social media filters. I guess it'd be great in the future just for people to come to a conclusion that there is a source that they can trust and rely on and whose perspectives they align with or at least are learning from.

I guess this sort of taps into the whole fake news issue that came up with the past election, but I guess — I hope where we're going is a place where Software Engineering Daily blossoms into sort of a more multifaceted media outlet and can maybe even compete with some of the bigger media channels. I hope they'll be more continued dialogue surrounding all news, but particularly within software and even the podcast industry and that people will have more discoverability, better truth in reporting, and also a better chance to share and have a conversation about those things in an intelligent way.

[0:27:47.1] PM: Exactly. I think Erika hit the nail on the head. I completely agree that the biggest problem is knowing who to trust or even which articles to trust, because with news and with journalism, there has always been this kind of the main task of the listener or the viewer to kind of take in the information that the newscaster is giving them and then form their opinion about it.

My experience or my understanding of this is that, previously, there wasn't that metacognition aspect of it where you had to also figure out whether the source that was providing you

information was legitimate or not. You could take the information and assuming it's true and then figureyou're your own perspective on that information.

Now, there is that extra load of having to figure out, "Hmm, is this source legitimate? Is this fake news? Should I try and get this verified by reading about it in another publication?" I think all of that causes a lot of mental fatigue when all I want to know is what is going on.

The fact that we are in a world where trust between publication and its viewership or listenership is eroding is really challenging. Add on to the fact that the publications are also being attacked from above by businesses and governments who have a vested interest in making sure that publications are neutered, then basically news outlets are being strangled from both sides. Then you add on to the fact that the Internet is making ad-base revenue completely change and forcing publications to change their business models and react to technological improvements and a different way of delivering news to a new generation, and you have this really uncertain ground that media as a whole is in right now.

The counterpoint to that is whenever there's uncertainty or a shift in the way things are working, there's a huge opportunity. I think our collective and especially y'all being journalists, the collective goal here is to make sure that that opportunity is not seized by unsavory groups. You obviously don't want that uncertainty to be exploited by Russian propaganda, right? That being, right now, the antagonist in vogue. This is not to comment on any of the stories about Russia or fake news, but the goal of us is responsible citizens, people who want to make sure that truth gets heard and seen is that we take these challenging environments and use them as a foundation an opportunity to create and to build back that trust and to get more viewership. Now, how to do that? That is something altogether that we can go into, but I think that's the biggest opportunity right there. Like Erica said, if SEDaily or whatever is SEDaily grows into being is the source of truth and people really believe that and they are assured of that, they will come back. That is how progress will be made.

[0:30:54.9] JM: What is the role of technologists here? Because as we've seen in different industries, whether we're talking about the mortgage trading industry, the derivatives trading industry I should say, or add tech, or any kind, like agricultural engineering. When engineers do not take a step back and look at the bigger picture for the tools that they're building, it tends to

make them myopic and it lets them get easily exploited by the people who are telling them what to do.

As engineers — I guess there's two different questions. One; there's a question of what are engineering companies liable towards? What are their responsibilities in this changing media environment? Then what is the role of the individual engineer working at one of these companies?

[0:31:52.4] PM: It's a tough question to answer because it's has multiple facets. I'll start out talking about the companies. I think the goal of any engineering company should be to make a great product to serve some user and to generate revenue. That may not always align with the moral principles of truth and reporting honestly, but I think the fundamental point at hand is what are the shared interests between all parties; the viewer of some sort of information content, the person who delivers that information content, and then the person who provides revenue, be it an advertiser or a sponsor of that content?

Because the landscape is evolving so quickly, I think there has to be some really innovative or more complex solutions to these problems. I was reading about something recently, it's called the BAT, or Basic Attention Token, and this is something that Brenden Ike and the brave browser team are working on.

The ideas basically you commoditize attention such that viewers or readers and advertisers are rewarded and their interests are aligned in making sure that when an advertiser pays for an ad, they are assured that people are actually viewing it. Then both the viewer and the content creator are rewarded or paid out in a proportionate amount to the viewership.

I'm totally butchering the concept of BAT because I'm only familiar with it on a surface level, but I love the idea that we're thinking about these tokens, or these coins built on the Ethereum blockchain to kind of align the interests of multiple parties. Whereas previously, you know Jeff, that there's a lot of misalignment of interests in the advertising industry leading to stuff like ad fraud or botnets to exploit naïve advertisers who'll just throw money at anybody and hope that their ads are viewed or listened to.

I didn't fully and your question, but I think the gist of what I'm trying to say is the world is changing in terms of how people access and consume content. To get to an equilibrium ground where things are done right, we will require some really creative solutions. I think people are starting to realize that and work on it. A lot of companies are still playing the game as it's always been played.

[0:34:29.6] JM: Erica, what are your thoughts? Do you have any thoughts on this? Pranay talked about the responsibility of the company. What I'm also curious about is the responsibility of the individual engineer. I've worked at companies where zooming out, where I take the bigger picture, I say, "Well, this situation that I'm working on, this product is — Or this feature that I'm working on is pretty questionable whether this is actually helping anybody. It's certainly helps the bottom line, but seems to exploit somebody else." Is there an ethical responsibility for the engineers there?

[0:35:09.0] EH: I mean there absolutely is, and that's just a question of making it in the modern world, like your day-to-day survival concerns and then stepping back and looking at the broad ethical repercussions. We both just recently read the Elon Musk biography, and one of the engineers in the book said one of our dilemmas is that the greatest minds of our generation are working to get more people to click ads.

As you've seen in some of your add tech discussions, I mean that's so true. I mean the bottom line for a lot of these companies is revenue and at what point on a day-to-day basis. With the engineers being so much in depth in their work, the feature they're working on within the product, at what point in the day do they get to zoom out and see the big Google maps perspective on the right or wrong of the situation?

I don't know. I think every individual probably has to make that decision for themselves on a day-to-day basis, reflecting on their work and is it ethical. Can they sleep at night with what they're doing? I would still hope that a lot of these companies are working toward a good product, like you are saying,helping people building a great and useful thing.

I guess we — We've talked to so many engineers on the show, so what have you felt about that?

[SPONSOR MESSAGE]

[0:36:47.5] JM: Ready to build your own stunning website? Go to wix.com and start for free! With Wix, you can choose from hundreds of beautiful, designer-made templates. Simply drag and drop to customize anything and everything. Add your text, images, videos and more. Wix makes it easy to get your stunning website looking exactly the way you want. Plus, your site is mobile optimized, so you'll look amazing on any device. Whatever you need a website for, Wix has you covered.

So, showcase your talents. Start that dev blog, detailing your latest projects. Grow your network with Wix apps made to work seamlessly with your site. Or, simply explore and share new ideas. You decide. Over one-hundred-million people choose Wix to create their website – what are you waiting for? Make yours happen today. It's easy and free. And when you're ready to upgrade, use the promo code SEDaily for a special SE Daily listener discount. Terms and conditions apply.

For more details, go to www.wWix.com/wix-lp/SEdaily. Create your stunning website today with Wix.com.

[INTERVIEW CONTINUED]

[0:38:13.2] JM: I think it's it depends on what situation you're in. If you're an engineer who doesn't have a lot of options and you're having a lot of trouble finding a job, then if somebody offers you a job where it's like, "Yeah, you're building a botnet that we're going to sell to ad agencies so that they can send bots to the campaigns that they're running and get fraudulent clicks." If you're desperate for money, hell yeah you're going to do that.

Then that's fine, because that's an interesting difficult engineering problem just like the same people — The people who tanked our economy in 2008. I think a lot of them were probably people that were just in the industry for a long time. They believed in the efficient market hypothesis, or they were fresh engineers out of school who were from physics backgrounds and

they just saw a really interesting system and they saw these beautiful market dynamics and they wanted to build something that interacted with that system.

As they saw the interactions evolve and they saw the money just pouring in, they said, “Let's amp this up. This is really interesting. This is working really well.” Then it took the economy tanking for people to realize, “Oh my God! We built a monster.” It's curiosity. It's human curiosity, and it just takes you to dark places if you don't regularly take a step back and examine what you're doing. We try to do that sometimes on the show. Some people hate it. Some people do — Speaking of it, people who write in. Some people stop talking about philosophy, stop talking about politics. I don't care about that crap. Go back to talking about JavaScript frameworks.

There's also the opposite response. Actually, there are many people who write in and say, “Oh my God! I loved that episode where you just talked about the nature of reality.” That was one of the most popular episodes was that show with Donald Hoffman, the nature of reality, and that was barely about engineering, but people loved it. On the other hand, we've done some shows about politics and we get some people writing in, they’re like, “Stop doing that.”

I think there's a balance in talking about this, but I think engineers really need to think about it, because there's just this narrative that is so toxic that, “Oh, you’re just an engineer. You just do what you're told and you just like to follow the orders, you follow the spec,” and this is something that engineers often propagate.

I talked to engineers who worked in ad tech, and I'm like, “Do you realize what you're doing?” They’re like, “Yeah, I’m just an engineer. I'm not responsible for this.” I’m like, “What are you talking about?”

[0:41:04.9] EH: Yeah, that’s a complicated issue for sure. Like you said, it's based on both the job market, maybe that individual’s perspectives and their goals for the future.

[0:41:17.3] JM: Yes, complicated issues. Anyway, let's talk about some other themes from Software Engineering Daily the we've covered. Pranay, the theme that you’ve become an expert

in is, I think, JavaScript, or probably the intersection between JavaScript and mobile development. What is going on at the intersection of those two areas?

[0:41:42.5] PM: I think, yeah, expert has definitely the mapped the right word. I think, as we all know, impostor syndrome is something that is an affliction for life as a software engineer. I have definitely drilled deeper into frontend JavaScript, not necessarily on mobile clients, but on browsers. I think the bigger theme is a standardization or convergence of the JavaScript community upon a set of tools that we all agree upon.

I think a lot of these can probably be attributed to TC39 which for those who don't know is the group that decides upon JavaScript standards and figures out what are the next set of features to be introduced with the next iteration and ECMAScript. I think this started largely with ES6 as a convergence point of people agreeing upon something, because prior to this, there were many disagreements and some people wanted to use Vanilla JavaScript, some people wanted to use Coffee Script, and people wanted to use TypeScript, and there all these tools floating around and a lot of these resulted in this phenomenon called JavaScript fatigue, where things are moving so fast, people couldn't seem to agree upon anything and web developers felt like, basically, the rug was being pulled out from under them every month couple of months. You would finally learn how Ember worked and then Angular would be released, and then you'd sigh and say, "Okay, one more time. I'll learn Angular." Then you learn Angular, and then React is the new hot stuff.

Then you're tired, you're exhausted, because as any sufficiently technical person knows, it takes time to become a master of your craft, to learn your tools well. I'm a strong believer that the specific tool you're working with doesn't really matter all that much provided it's not like some super junkie tool. If you have a good tool, a lot of the benefits will be derived from you becoming better and more skilled with that tool. If the rest of the community is deprecating your tool every two months, you're going to get exhausted because you're not given the time and the mental freedom and the joy of becoming better with any one thing.

I think that's been the biggest change, and it's a philosophical one that we can all agree upon certain tools to use and allow for ourselves time to become good with those tools so we can figure out, "Hey, what the flaws with these tools?" and then work on a kind of more methodical

and patient and kind of relaxed way of improving those tools to accomplish our goals. That's been really a joy for me, because I feel like I got back into development with JavaScript at a time when things are kind of settling down.

[0:44:32.6] JM: Right, there are some convergence for better or worse on React. What about React — Stuff that feels futuristic to me. React Native, certainly feels like — Where is React Native going to stop? It seems like it's going to keep going and it's going to get really performant and maybe it'll become the cross-platform tool of choice.

[0:44:59.9] PM: I can't say for sure, but I think one really interesting anecdote here is that John Carmack who works on Oculus and VR at Facebook posted a couple of tweets recently where he initially — Where he mentioned that, initially, he was skeptical of React VR, which is a React-base library to build VR interfaces.

He was initially skeptical that this would be the right approach to help developers democratize VR development. After he saw developers working with an undisputed iteration and how quickly they've picked it up, given that they were familiar with React from the front-end or from React Native, he changed his mind. He thought, "Wow! This is actually a more powerful tool than I could have understood."

I'm not sure exactly what the reason for that is. I think, obviously, React has some intrinsic principles that are easier to pick up on or easier to reason about. It's functional paradigm of programming which is often preferable to object-oriented and is in vogue right now.

I think, also — I think my personal opinion is that it, again, goes the point that familiarity is a huge boon for development. If you can take something that you worked on one domain and ported to another, that cross domain sharing of knowledge is really knocks down the barriers to getting started enrolling on something else.

Since I am at a sufficient depth in React right now, I feel confident that I can go into VR development and at least know a little bit about what I'm working on tomorrow. I think that leads to a lot of developers being interested in React.

[0:46:40.8] JM: Right. The ability to go from platform to platform as we get to augmented reality and virtual reality and developing with a TV. We've talked about fatigue. Speaking of fatigue, people are really burnt out from this disaster that we had with the mobile dichotomy, I guess, between android and iOS. There's a whole generation of programmers who just didn't want to get involved with mobile development because they're just like, "Yeah, I'm just a web developer. I can't do all of these stuff," and there's no reason it had to be that way. It should have just been one language.

If we'd had some sort of consensus across the world, maybe it was good to have this long period of pain and lack of consensus and now we're just like, "Oh my God! Let's converge, please, on something."

Pranay, have you looked into web assembly at all?

[0:47:30.3] PM: I've read a little bit about it, but I think as a JavaScript guy, I'm basically crossing my fingers and hoping it doesn't knock me out of a job.

[0:47:39.0] JM: Okay. Right. You're saying —

[0:47:41.8] PM: I think web assembly is actually pretty cool. I want to spend some more time digging into it, but from what I've read and understood, it's still a ways away from practical use cases. But it's definitely — Mozilla has done a fantastic job with it and I think people are really excited about unlocking different languages on the client side because even though JavaScript is standardizing, for some people it leaves a sour taste in their mouth. I think a lot of that is due to the initial versions of JavaScript.

I think, generally, if there's more freedom and more tools to work with, despite what I said earlier about standardizing on tools, I think consumer choice and developer choice is very fundamental because a developer should be happy, and if they're not happy with what they're working with, they should be allowed to change.

[0:48:32.5] JM: Yeah. All right. I want to wrap up by talking about where you both think we're going and what your plans are for the future, because I think — Believe it or not, I think the

listeners are curious about the back story of Software Engineering Daily, where things are going and the people who are involved. Why don't you just both — Let's close off by talk about what you think is changing in the world and how that is informing your goals for the future. Erikca, why don't you go first?

[0:49:08.7] EH: Well, I think what Pranay was talking about with cross-platform development is incredibly important. I feel like we're way behind on this. I hope to see a lot more of that. How that will apply to Software Engineering Daily is that's we'll see better mobile interaction, better IoT applications, discoverability of content.

As far as the business goes, I'd love — And we've talked about this quite a bit, Jeff. I'd love to see us grow into a more multifaceted media business. This year, we rolled out our first video on microservices, and that was really fun. I think we can either grow into a podcast network or perhaps put on some remote conferences or even get into the e-book or audiobook realm or grow into more of a news publication. There so many options there, and I think we'll just have to see what our capacity is, what people we could bring on to help with that and formulate a plan about our own peak. There's so much opportunity right now within the media and a lot of great tech outlets and inspirational figures to draw inspiration from. I'm really excited about where we're going.

[0:50:24.9] JM: Great. Pranay?

[0:50:26.2] PM: Jeff, I definitely want to hear your perspective on this as well, because I think SEDaily, like Erika said, has a lot of opportunity right now especially to diversify from just being a podcast and capture more of the generalist market for tech content and tech media.

There's recently some mainstream media outlet, New York Times, or whatever covered a piece on uBeam, and the former vice president or some guy who was a higher up in engineering at uBeam laughed off the article as being basically gullible journalists just buying into something that a teach company said.

I think SEDaily is really positioned and has the opportunity to be that voice that can call bullshit and say, “No, that’s not true, because I'm an engineer and I know that’s how it works, so give me the truth.”

I am totally excited about that becoming — First of all, because of technology eating everything, that becoming more of a product that is demanded and also something that more people are going to be technically skilled and be able to devour content at a higher level of technicality. I’m super stoked about that and want to hear your perspective for closing thoughts.

My perspective on the world is that I think we’re at a really kind of weird time of foment that I think the current events over the last half a year, a year, have, I think, beyond just me have confused a lot of people and I think it just owes to the point that our technological sophistication is growing faster than our ability to keep up, and our old models of the economy and about employment and about how people derive and provide value in the marketplace is being undermined.

This kind of rude awakening is really destroying a lot of lives to be candid. A lot of people can’t get employment anymore. A lot of people can’t retrain and become more technical to keep up with complex service level job requirements, and I have a lot of compassion for the people who are suffering from this because it's easy to look at things from a black-and-white technical perspective and say, “Yeah, we’re becoming more advanced. More people have longer lifespans. More people have toasters and microwaves,” but a lot of people are suffering because they don't have a purpose that aren’t able to provide a value to the economy and provide for their families.

I think our collective responsibilities is to figure a solution to that whether that is the basic income, like Zuckerberg and other people have posited or some other different form of vocational training. Whatever that is, we have to come to a consensus as technologists beyond just looking at our technical work and saying, “Gee! That’s interesting. We need to think about their larger impacts on people and have some compassion for what the average person is going through.” It’s a vague thing to say, but because it’s a big open question that I think entrepreneurs should be very excited about addressing. It's a very scary but also exciting time for the world.

[0:53:39.7] JM: Well, talking there, you just reminded me of all the conversations that we had when you were a full-time or a part of Software Engineering Daily, and I think we share a vision for more technical content that takes the consumer of the content seriously and also brings a level of technical seriousness.

I think what you and I both shudder in response to is the type of writing that treats the reader is an idiot or that's makes the journalists is comfortably naïve. I mean you talked about the uBeam example. That just happens with people who are generalists, reporters. It's almost unavoidable, because there's only so many journalists in the world and you can't be an expert in everything.

A lot of shows I've covered, like I listened back to them, like, "I did a terrible. I basically presented misinformation." You do the best you can. You're talking about the uBeam example, what I was thinking about was there was this story about the meth bot story, like, "Oh, we finally discovered ad tech with the meth bot story." I don't know if guys saw that, but this is a big store in the New York Times about a botnet that was making millions of dollars a day, and it was a puff piece for White Ops which is an anti-ad fraud. It's a bot detection company and it's just like a puff piece for this company and New York Times can't identify that. They can't realize that they're being duped into writing a puff piece about an anti-add ad fraud company and the whole ad tech stuff, like this is — Honestly, strangely, this is one thing that really motivated me to realize, "Oh my gosh! There's so much opportunity for oversight in the technology world because the ad tech is just this big ball of fraud. It's a big Wall Street 2.0 with no regulation and brands are losing tons and tons of money and fraudulent sketchy people are wasting their time building systems that are doing nothing but serving ads the bots, ultimately."

It's so easy for me to get riled up about it, and I feel like that's a good sign that there's something here. There is a glut of misinformation, and I think the broader case is that there's a glut of misinformation around technology because the people who should be reporting on it can't report on it and the technologies that are building these things have learned that that they can bluff the journalists.

[0:56:26.0] PM: I think it's Emperor's new clothes all over again, and somebody's got to yell out, "The Emperor is naked!." I think that would be so amazing if SEDaily can fill that role.

[0:56:39.9] JM: Yeah, the thing is it's not just software engineering. You and I talked a lot about — You studied chemical engineering, and we would talk about chemistry, and you would always tell me stuff about chemistry that blew my mind and would just like scare the crap out of me. It's just like —

[0:56:54.5] PM: You know I'm excessively paranoid about that, but I agree.

[0:56:58.4] JM: There's actually no way to know if you're excessively paranoid, because we don't have the science to disambiguate what are the causes of diseases in our world.

[0:57:10.3] PM: Right. I think some amount of litmus is necessary and I think that's the point that you're making that nobody can ever know for sure. Like you said in previous episodes, you may have unwillingly disseminated misinformation. I think the idea is to hold yourself to a really high standard of reporting, and that requires training yourself, becoming more technical and working hard to make sure that you revise past missed information that you constantly keep up to date, and that is challenging for folks who don't have the technical foundation, but you and Erika do. I think you guys are doing a great job. The keys, as the audience keeps growing, as the formats keep changing, that you guys hold to those core principles of honesty and working hard to keep up-to-date with this rapidly changing technical landscape and knowing that every step of the way you're going to be shining beacon for technologists and increasingly general audiences in reporting on the technical world and to be able to call truth where you see truth and lies where you see lies.

[0:58:24.1] JM: Let's hope so. I'm going to let both of you get back to your beautiful Sunday. Pranay, it was really really good to catch up with you, and I look forward to seeing you more in the future. I hope you come back to San Francisco soon.

[0:58:39.4] PM: I hope so too. I look forward to working with you, and with you, Erika in the future. I mean this is — Like I strongly believe, this is super exciting time and there's a lot of cool stuff to work on.

[0:58:52.4] JM: Great. Thanks Pranay, thanks Erika.

[0:58:54.3] EH: Thank you both. It's been great working with you.

[0:58:57.2] PM: Awesome.

[0:58:57.5] JM: Great.

[END OF INTERVIEW]

[0:59:01.5] JM: Gatsby is a platform for improving your conversion. Let's say you were re selling cat food on your website, you're doing a e-commerce site for cat food. If you want to run a discount for that cat food, you can use Gatsby to embed a widget for that discount. It will say 15% off, and if people click on that, not only do they get 15% off but you get to gather information about who clicked on that discount. You can also use Gatsby to embed social media widgets that look really nice. It's really nice design.

The reason we're talking about this on Software Engineering Daily is because this is the Gatsby API. You can get access to the Gatsby API by contacting api@thinkgatsby.com. Whether you're selling a SAS product or you're selling cat food, Gatsby really useful for doing things like developing custom in-house promotions or connecting the user data of the Gatsby app to specific endpoints or integrating Gatsby with your other in-house software. You can do A-B testing for multiple promotion campaigns. It's really flexible API for different types of marketing. You contact api@thinkgatsby.com to find out more. If you're not interested in the API but you're just interested in the Gatsby product itself, go to thinkgatsby.com.

[END]