

EPISODE 376

[INTRODUCTION]

[0:00:00.5] JM: Software engineers have a skill that can be applied to solve problems outside of a code base in real life. Analytical skills can be used to evaluate investment opportunities. Creative thinking can be used to build businesses. Communication skills can be used to build and enhance relationships.

John Sonmez is a software engineer who created the Simple Programmer, a community of developers who discuss strategies around software, business, and life. He join me on the show to discuss these topics and others as well as his new book, *The Complete Software Developers Career Guide*. John is the author of *Soft Skills*, which is a popular book that engineers often purchase when they need to develop those soft skills and I highly recommend an episode that he did on Software Engineering Radio about developing soft skills. It's a classic, and I'll put it in the show notes.

By the way, Software Engineering Daily is looking for sponsors for Q3. If your company has a product or service or if you're hiring, Software Engineering Daily reaches 24,000 developers listening daily. Send me an email, jeff@softwareengineeringdaily.com if you're interested in sponsoring the show. Now, let's get on with this episode with John Sonmez.

[SPONSOR MESSAGE]

[0:01:29.0] JM: Hosting this podcast is my full-time job, but I love to build software. I'm constantly writing down ideas for products; the user experience designs, the software architecture, and even the pricing models. Of course, someone needs to write the actual code for these products that I think about. For building and scaling my software products I use Toptal.

Toptal is the best place to find reasonably priced, extremely talented software engineers to build your projects from scratch or to skill your workforce. Get started today and receive a free pair of Apple AirPods after your first 20 hours of work by signing up at toptal.com/sedaily. There is none of the frustration of interviewing freelancers who aren't suitable for the job. 90% of Toptal clients

hire the first expert that they're introduced to. The average time to getting matched with the top developer is less than 24 hours, so you can get your project started quickly. Go to toptal.com/sedaily to start working with an engineer who can build your project or who can add to the workforce of the company that you work at.

I've used Toptal to build three separate engineering projects and I genuinely love the product. Also, as a special offer to Software Engineering Daily listeners, Toptal will be sending out a pair of Apple AirPods to everyone who signs up through our link and engages in a minimum of 20 work hours. To check it out and start putting your ideas into action, go to toptal.com/sedaily.

If you're an engineer looking for freelance work, I also recommend Toptal. All of the developers I've worked with have been very happy with the platform. Thanks to Toptal for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:31.5] JM: John Sonmez is an author, a teacher, and entrepreneur. John, welcome to Software Engineering Daily.

[0:03:37.7] JS: Thanks. I'm glad to be here.

[0:03:38.9] JM: You write a lot about the softer skills that an engineer needs to develop and should develop in addition to a lot of the technical content that you produce, but I want to start off talking about the broader subject of skill development because it's something you spent a lot of time thinking about. What are the skills that software engineers frequently don't develop as much as they should?

[0:04:06.3] JS: Let's see. We could talk about definitely from free different angles. I would say foremost, the soft skills, but I kind of get the impression you're looking for not to go down the soft skills road just yet. Is that right?

[0:04:22.4] JM: I think you could go down either route. I'm most curious, because you see people in different places in their career arc. You probably see people early on and in the middle

of their career and later on in their career. There's probably some themes whether they are technical skills or softer skills that you say, "Wow! If this person would have just knocked it out earlier on their career they would've compounded much more interest throughout their career." I think just the skills that are the most valuable yet get underdeveloped.

[0:04:55.5] JS: Okay. Yeah. I would say you definitely by far and large the biggest one that's most valuable that will have the largest compound interest for, I think, any person, not just a software developer over the course of their life, is the ability to be able to teach themselves, self-education. That's one of the biggest things because I would probably rank some of soft skills type of things after that, but if you have the ability to teach yourself, if you have that ability of self-education, then everything becomes accessible to you. All the doors opened up to you and you no longer rely on what other people tell you. You no longer rely on the system in order to educate yourself and you can go in any direction that you want.

I think that's probably the most valuable skill, and it's surprisingly underdeveloped. A lot of developers are self-taught, but they still don't really have the skill — They still don't really have a really good handle on how to teach themselves new things quickly, how to learn quickly, how to absorb information. A lot of the ways that we tend to learn is we tend to read a book cover to cover or we go and enroll in a class and that's usually not the most efficient way to learn and to self-educate yourself and it requires you to have a teacher. It requires you to go down a traditional path. I try to teach a lot of developers how to quickly pick up information. If you want to learn something, how are you going to learn it as efficiently as possible? I think that's something that I found that has been extremely valuable not just in my development career but in my life in general.

[0:06:42.8] JM: I like that choice because that's a recursive skill. If you're learning how to learn then you're going to learn the other subsequent skills that are going to help you more quickly. I also think it's infinitely difficult thing to do. I certainly check myself on a regular basis and say, "Oh, I'm making some sort of a mental bias that's keeping me from learning as quickly as I should be doing." I think it's that's a skill that one never stops developing.

[0:07:18.8] JS: Yeah, definitely. It's a lifetime skill. Just like learning itself, it's a lifetime skill and you get better at it as you figure out. You're working with a very small amount of available

memory that's accessible of RAM in your head. Even your retrieval processes is not as efficient as you'd like it to be, so you've got to figure out really how to learn things where you don't have to know everything but you know enough and you can navigate enough to find that the pieces that you want. Almost like making — One of the ways I approached you today is utilizing what have. Using Google as an extension of my brain, right because I don't need to know a bunch of stuff that I can look up easily but I do need to know what I need to look up and that's really the key today.

Back in the day before we had our brains connected to the internet and all of the available information that we had, we had to memorize stuff and it had to be in our heads, Now, we can almost index into — We need an index in our head, not the actual data.

[0:08:35.0] JM: What I like about your material is you have focus on both the short-term and the long-term thinking. I think there's a lot that has been written and content that's been produced about short-term tactics for improvement, and that stuff is always valuable. As somebody who's built a lot of different angles to your career that are kind of building and evolving in different ways, I can sense that there is an attitude of long-term thinking to how you have built your career to where it is today. How much long-term thinking should an engineer do and how does long-term thinking factor into creating and crafting a career that a software engineer will enjoy for the duration of their life?

[0:09:29.0] JS: That's a great question. That's a question everyone should be asking themselves, I think because that is one of the questions of life, really, of your career. I've thought about this a lot. Really, where my thinking on this comes from is from two places from computer science algorithms, but more so from real-time strategy games. I used to play resource management games, like StarCraft back in the day, and World of Warcraft and all that.

Those games where you have to manage resources, if you've played one of those games, you know that the best strategy that you could possibly do in the game — The best way you could invest resources is long-term. In StarCraft, you have those big Protoss carrier ships that — Either the biggest units that cost the most, the relative cost to the damage they did was so great. They did so much more damage, but they cost a lot so you'd have to sacrifice resource up front.

I think of life in your career in the same way. You can think of it in finances. You can think of time you invest. If you could optimize for long term only that would be the best thing possible, but in the meantime you got to pay the rent, you got to eat. You got to take care of other things that require your time. You do have to do some short-term thinking in order to survive long enough, in order to benefit from the long-term.

The optimal strategy, or optimal solution to this problem is to make sure that you're setting yourself up for the long term as much as possible with one caveat which is that you have to live long enough or you have to have enough of the time that you actually benefit from the long-term for it to be worth the investment.

For example, if we looked at this from a financial perspective, the best way you could possibly invest your money to make the most money financially would be long-term. If you just looked at something like, let's say, CDs or most of the investment vehicles that you're locking up your money for the long-term, that's going to have the highest yield.

Let's say that I gave you some kind of investment or I said, "Okay, if I crated this crazy CD," certificate of deposit, and I said, "Okay, what I'm going to do is if you lock up your money for the next 50 years and you can't touch it, then I'm going to give you a 3000% return on your money per year," let's say. That's a ridiculous amount, right? It doesn't matter, because you can spend that money because you'd be dead.

You got to find that optimal point, and I think that's such in life, and that's why I'm actually — Not to side tangent here, but I'm against 401(k) and retirement plans because my philosophy and my idea of life is, "You know what? I really don't care when I'm 65." If I want to make it now and enjoy the fruits of my labor in an earlier time period — It doesn't mean that I'm not investing for the long term, but I'm thinking a little bit differently, not that long.

Again, if we go back to the career, if you look at different stages of life like, for me, I don't bill by the hour as much as I can avoid as possible. If I bill by the hour now, I bill a minimum of \$2,000 an hour as my hourly rate. I know that sounds ridiculous and you say, "Most people aren't going to pay that." Yeah, but some people do, and if they're willing to pay that then I'm willing to make

a short-term decision. Other than that, almost everything else I work on in my career, in my life, is long-term. Even recording this podcast, it's going to go out there, it's going to last a while. All these stuff that I build, this stuff that I worked on, because I know that that's going to have a bigger payoff and benefit down in the long run. I'm building those carrier ships in Starcraft. I know that I'm going to investment my resources best that way. I'd I say to try to tie it to something practical.

As a developer starting out or as a developer considering these things, you want to think about spending as much time as possible long-term and just getting by as much as possible in the short-term. What can you do? Can you cut your expenses as much as possible? Can you emphasize learning and developing your skillset more than making the salary now? Can you invest in marketing yourself, building a blog, doing podcast, anything that's going to have that long-term payoff, maybe YouTube videos and tutorials, all of those things. A lot of times, that's why advise a lot of developers, I say, "Hey, look. Only Work 40 hours for your employer and spend the first two hours or first hour of your day creating something long-term for yourself; building your blog, creating a project, creating a business, something like that because that's going to have the bigger payoff in the long run."

This is a fantastic question that — We could talk about this all day because there's so much you could think about to optimize it, but it really just comes down to — If I laid it out to one simple principle, it'd be optimize as much as possible for the long term, for the near long-term, not the too too far long-term.

[SPONSOR MESSAGE]

[0:14:55.4] JM: Simplify continuous delivery GoCD, the on-premise open-source continuous delivery tool by ThoughtWorks. With GoCD, you can easily model complex deployment workflows using pipelines and you can visualize them end-to-end with its value stream map. You get complete visibility into and control of your company's deployments.

At gocd.io/sedaily, you can find out how to bring continuous delivery to your teams. Say goodbye to deployment panic and hello to consistent, predictable deliveries. Visit gocd.io/

sedaily to learn more about GoCD. Commercial support and enterprise add-ons, including disaster recovery, are available.

Thank you to GoCD and thank you to ThoughtWorks. I'm a huge fan of ThoughtWorks and their products including GoCD, and we're fans of continuous delivery. Check out gocd.io/sedaily.

[INTERVIEW CONTINUED]

[0:15:55.7] JM: Let's say I'm either — I think there's two broad categories of people who listen to this show, and one is developers who are just starting out or they're doing a coding boot camp. They're switching from another career into software engineer, and the other category is somebody who's been an engineer for a while. There's also a subset of people who've already started a business, so they're entrepreneurial. They're very happy with what they're doing. For those people, this may not apply as much.

For the people who are early on in their career or they are a little bit advanced in their career and they're hearing this, "Okay, I can do stuff on the side. I'll only allocate 40 hours to my day-to-day employer. Why should I allocate only 40 hours to my day-to-day employer and spend the rest of my time doing other stuff, like building my "personal brand" rather than doubling down on my career growth? Why wouldn't I focus on the corporate job and go deep on the corporate job and try to really do well in that kind of traditional position?"

[0:17:03.5] JS: For a couple of reasons. One of them, it's a fallacy to believe that putting in more hours for a corporate job will actually benefit you or get you to raise a corporate ladder. In fact, it may show that you're somewhat of a patsy and it may cause you to be stepped on and utilized and used, I would say.

In my whole career as a software developer, I am not saying that I never did put in the time and work extra hours, but from a large portion of it, in the latter half of it as a software developer, I limited my work to 40 hours even in places where people were working 60 to 70 hours, and not only did I not get fired, I got promoted more and I got paid more and I was the highest-paid developer because I thought my time was valuable and so did other people. Mostly people —

It's just a good life lesson, is that most people will think of you what you think of you, what you honestly think you, not what you pretend.

If you honestly think your time is valuable and you think you're a valuable person, there's a reason why other people aren't going to think the same of that, and that's just a general idea. To be more specific here, I would say that you're taking a huge, huge gamble if you're going to invest in one company, invest a huge amount of your time in one company thinking you're going to climb the corporate ladder. Not only are you taking a huge gamble, but doing that — That kind of thinking has what I call bounded upside, but really, you want to aim for things that if you're going to invest a lot of time in things, if you're going to invest in things, you want to aim for things that have unbounded upsides.

For example, it doesn't matter how good you are or how many hours you put in and how much you climb the corporate ladder, you're going to hit a ceiling. You're going to hit a glass ceiling at some point and you're going to be stopped. You're never going to get paid as a developer a salary of — I don't know. Say, \$500,000, or a million dollars, probably not, unless inflation goes well, but based on today's numbers.

If you're an entrepreneur, if you start your own business, if you have other income coming in, if you've invested in that way, there is no cap. It's an unbounded upside thing. Now, the chances of you becoming a Mark Zuckerberg and making billions and billions of dollars, very low, but you've got a pretty high probability of something that could actually result in a much bigger score in the long run.

I think a lot of developers to — Times have changed. Our parents have told us go to school, go to college, get good grades, and then go get a good job and climb the corporate ladder. That's not true anymore, because right now there's not as much stability and working for a single employer as there was. The economy has changed. Look at the gig economy. Look at Uber, and look at all these displacements going on in places where what are taxi drivers thinking right now. A lot of unions are really in trouble because there's no security of regular jobs.

If you go and invest all your time, if you're working 60, 70 hour weeks for some company, whether it'd be a corporate job or a startup, which is even worse, because that's just a lottery

ticket and there's really no guaranteed payoff there at all. If you spend all that time and you're investing and building someone else's empire and doing — Honestly, the other piece of this too is that those hours don't benefit you very much as far as your own personal development because there's a difference between just working hours and deliberate practice.

What I'm saying is, "Hey, look. You invest all that time. You may not get a payoff. You may get laid off. It's not going to carryover necessarily to another job, so it's a pretty bad bet," but take those extra hours. If you're going to work 60 hour weeks, I'm all for it. Hell, I work 70 hour weeks most weeks, probably more than that a lot of my career, but I took those extra 20 or 30 hours per week and I put that to work for myself.

Sometimes I was building a business, but sometimes it was just reading a lot and investing in my career or learning new programming language on my and just enhancing my skills because that has a payoff that exists outside of that particular entity or company. It's very very limited and short-term thinking to think I'm going to climb the corporate ladder at some company or I'm going to kill it with this startup and they're going to promote me up. That's not necessarily true and those things don't carryover. I'd rather invest — Again, we talked about investing in the long-term thinking, investing 20 hours extra at your company to hopefully climb the corporate ladder is short-term thinking. Investing the extra 20 hours in your career learning a new programming languages, learning a new skill, building a business, building a blog, marketing yourself, any of those things, that's long-term thinking.

[0:22:04.6] JM: Right, and I think what you're saying is it's more about allocating this time to — You say, I'm going to work 70 hours a week, 40 hours for my day-to-day job, 30 hours for myself, and then under that 30 hour arc of working for yourself, you could do any variety of things. Whether you are starting a business, you have a long-term view for that business right now for that side business, or you're just reading and kind of sitting low and just kind of thinking about opportunities, that can be really valuable.

You made a lot of money teaching courses on Pluralsight and it was an opportunity that surprised you. As far as I understand, you kind of stumbled on to that and was like, "Whoa! This is actually really valuable." How do you look at the idea of opportunism, because that was — Okay, this opportunity fell onto your lap and you're like, "Okay, I got to pounce on this." It also

seems like that was something that you cultivated by being ready for that opportunity to fall into your life.

For the type of person who they spend 40 hours week at work and they spend 30 hours like hacking on stuff and learning stuff and reading stuff and they want to start a company or they want to build a software product but they don't feel like they're ready right now or they're kind of like waiting for the opportunity to fall into their lap, is there a way to cultivate that opportunism and be ready for the opportunity when it comes to you?

[0:23:34.0] JS: Oh, sure. Yeah, definitely. This is one of those things where I get into heated debates with people about because they say, "Well, you got lucky, or people get lucky or successful people get luckier. You luck in the equation." I totally agree you need luck in the equation, but here's the thing that I would say is that we all have these stellar opportunities that were going to get like so many of them in the course of her life, but 90% of them we won't recognize, and then the ones that we do recognize, like 90% of those, we won't be prepared for.

Then even of those that we do recognize and we are prepared for, I would say — I'm just making up percentages here, but I'd say 90% of those we don't follow through enough to actually benefit, and that's the key like. In order to really be successful, in order for opportunity to hit you write, it's not enough just to be prepared. It's not enough to just recognize it or to have it come to you. You've also got to have to follow through in commitments.

Success, really, in my mind comes down to these three things. One; it's the commitment. Two; it's the persistence. Three; it's the ability to work without motivation. To go on without focusing on the results when you don't feel like doing something.

If you look at, for me, for example, at the Pluralsight, what happened there? There was an opportunity that happened but I had been working. I've been building a blog. I've been marketing myself. I've been putting my name out there. Working on developing my skills, a bunch of different skills, and so it made it more likely that I would have a connection that had offered me the Pluralsight opportunity. When I had that Pluralsight opportunity, that wasn't just like, "Okay. Now, here you go. Now, you can just be a millionaire." There was a lot of other authors that were doing Pluralsight courses, but they didn't do as many courses that I did. I saw

the opportunity and I did 55 courses over the next couple of years, and that made the difference, but I was prepared to be able to do those.

Again, it required some luck and some opportunity for that to happen, but I get lucky all the time. It happens to me all the time. I find these lucky opportunities, but most the time what I found is that we don't see them. I could've easily not seen this one or could've easily passed me by. I could've not done a course or I could have just done one course and I wouldn't have seen the opportunity, or not even had the opportunity to show up at all because I wasn't doing the other things to market myself.

What I would say is it comes down to this, is you got to get up to bat all the time and you take a ton of swings and eventually you're going to connect, you're going to hit as long as you're willing to keep on swinging and swinging and swinging. A lot of people are waiting. They're waiting for that perfect pitch to come and they'll take a swing, but when that perfect pitch comes and they're like, "Oh, yeah. This is the perfect pitch." They go and they go to take that swing, they realize that they don't have the muscular strength to actually swing the bat because they've never swung it before.

If you're not willing to swing the bat and take a bunch of misses, you're not to get it. It's just not going to lineup. It's a numbers game, and so you've got to be taking enough swings. You got to have enough practice to be able to hit the pitch when you actually do, and then when you start hitting pitches, you got to keep on hitting that pitch. You got a run the full bases. You can run to first base and then be out of breath and be like, "Oh, man! This is too hard. I'm too tired. Okay, that's enough." You got to be able to do all those things if you want it to lineup.

I think there's a side of it too where you can't force things. I think you can't force opportunities, but it's just a matter of training. You train, you train, you train, and then the one day the enemy shows up at your doorstep and you're ready. When the enemy shows up at your doorstep is not the time to start training. It's too late at that point. It requires all those things to be in alignment and most people, honestly, aren't willing to put in the effort that's required. They just want ne opportunity, but they miss the fact that — They're going to have so many opportunities in their life. Most of them they're not going to be prepared for and they're probably not going to follow through on them.

[0:28:02.1] JM: Hearing you talk about this, it kind of makes me understand the simple programmer brand, the John Sonmez brand which is maybe you started out just thinking, “Okay, I’m going to do some education stuff around career development and some education stuff around how do you build a business with,” but that leads to, “Okay, how do you keep yourself prepared,” and then that leads to, “Okay, here are some lifestyle suggestions,” and you have these principles around stoicism and health and exercise and pretty soon you’re talking a whole way of living that.

There’s a lot of podcasts and blogs and material around these topics of stoicism and health and exercise and you got people like Tim Ferris and Ryan Holiday, all these other guys who kind of built personal brands around this. I think this is a really good thing because a lot of these different people portray these well-proven principles that are not controversial, they’re not secretive. It’s the same stuff that Tony Robbins has been talking about for a long time. Do you see yourself as unique in the sense that you’re portraying these principles through the lens of a programmer? Am I portraying your vision and your brand correctly?

[0:29:27.8] JS: Yeah, I think you are. I would equate myself. Sometimes I tell people when they ask — When they ask what I do, it’s funny, I used to say I’m kind of like Tony Robbins for software developer, but I find not enough people really understand who Tony Robbins was. They thought, “Oh! He’s —” I don’t know what they think that he does, but you just do motivational speaking. He coaches every area of your life, and that’s kind of what I do.

Then I started saying, “I help software developers — I teach software developers how to be cool,” and that’s where I’ve landed, is that’s what I tell people most of the time now. It’s kind of joking, but it’s personal development. It’s kind of funny.

My lens for it is definitely different in the fact that I come from an analytical and software development mind. I think that that is unique to it, because for a couple reasons. One of them is that I’m a skeptic — I’m into the most woo-woo kind of crazy stuff that people are like, “Man! John, really? Seriously? The universe tapped you on the shoulder of John? Come on!”

[0:30:34.1] JM: [inaudible 0:30:34.0] of that.

[0:30:37.2] **JS:** I read books like *The Power of Now*, which is somewhat kind of mystical, like —

[0:30:45.7] **JM:** Eckhart Tolle, or what is it? Eckhart Tolle?

[0:30:47.0] **JS:** Yeah, exactly. I talk these kind of things that are out there; affirmations, these kind of things that. I came from an analytical — I was so skeptical of all that stuff and I started applying these things in my life. My path to it really came from I would talk to — I had the opportunity to talk a lot of millionaires and some decamillionaires and even couple of billionaires and I would ask them, I would say, “Hey, what is the number one book that you’d recommend that everyone read?” I have to ask this question. Whenever I find some super, super successful, I assume they’re not lucky. I assume that luck had a part of it, but I assume that they did those other two things I talked about.

It kept coming up this stupid book, this dumb book called *Think and Grow Rich* by Napoleon Hill and I’d read this book before and I was like, “Oh, brother! Give me a break. This is ridiculous. This is ridiculous stuff,” but they kept on saying it. I was like, “All right. You know what? If all these guys —” Here’s a problem with ignorance, if you’re ignorant, you don’t know you’re ignorant. It’s impossible to detect, if you think about that. If you’re ignorant, you can never be aware of your own ignorance. It’s not possible, by definition.

I’ve sort of developed this philosophy in life that says, “Okay, sometimes when I think I know it all, when I think I’m smart, when I think I’m wise, I have to look at people — And I think other people are crazy, but they’re more successful than me in some area or they’re super successful. I have to kind of say, “Well, is it possible that they’re wiser than I am? That I’m actually ignorant and that my analysis and everything that I’ve got going for me is — That they’re at some superior level and I can’t detect that. It’d just be undetectable. The ignorance is undetectable. Should I just give what they say if they’re qualified the benefit of the doubt and try it out?”

That’s sort of what I did there, is I reread the book with this new mindset and I’m still was a little skeptical, but I started looking at some of the other books they are recommending. Again, a lot of kind of crazy stuff out there, and I started applying that stuff in my life and, man, I started

getting a lot more successful, not only a lot more successful, a lot more fulfilled and happy in my life and it felt a lot better. I was like, “Wow! This is really — I got to reevaluate my position.”

That sort of led into the whole thing. That's where I sort of shifted and I still have that analytical mind. I'm still very skeptic at heart even though sometimes it doesn't seem like I am. I approached these personal development things not from a blind follower type of viewpoint, but from one who recognizes both that there're things we do understand and we need to analyze and scrutinizes those things, and there are some things that we don't understand but they just work and maybe some science behind them or maybe the universes is way more complicated than we think, but these things work. Let's just take some of these things on faith and try these things out even though that sounds really weird to say that.

That's where I come from. I say there's one other thing that I bring that's unique, I think, to this space, which is I was just doing a video on the this the other day, and I record two to three videos a day and I record — My YouTube channel is almost like a diary for me. A lot of people, a lot of the success gurus and stuff that you see out there, you see their success, you see their highlight reel and you don't see how they got there and you don't know if they're pulling the wool over your eyes and stuff, but I go and I go run a marathon and I film it and everyone knows that I'm working out in lifting weights and going for 40 miles a week of running and you see me on video and you see that I'm ripped with a six pack all your around. I'm not bullshitting you when I'm saying this stuff. You see how many hours I'm putting in. I'm talking about these things. Even just from YouTube, you see every day two to three new YouTube videos come out for several years now.

When you say, “Oh, well, you got lucky to get 100,000 subscribers on YouTube.” No. No, I didn't. I showed you exactly the process that I'm following. I think that's the piece that I try to bring that's unique is to say, “Look, not only am I telling — I'm not just telling you all these mantras for success. You're seeing me do it and you're seeing the results of it as you're watching me grow and I'm just giving you everything that I'm learning, my mistakes, my successes, my failures, all that stuff, and you're seeing it on a daily basis. There's no question. It's like reality TV. You can't say this didn't happen because you're seeing it. That's sort of the unique kind of approach that I take to it.

[SPONSOR MESSAGE]

[0:35:44.7] JM: VividCortex is the best way to improve your database performance, efficiency, and uptime. It's a cloud-hosted monitoring platform that eliminates your most critical visibility gap, providing insights at 1-second granularity into production database workload and query performance. It measures the execution and resource consumption of every statement and transaction, so you can proactively fix future database issues before they impact customers.

To learn more, visit vividcortex.com/sedaily and find out why companies like Github, DigitalOcean, and Yelp all use VividCortex to see deeper into their database performance. Learn more at vividcortex.com/sedaily, and get started today with VividCortex.

[INTERVIEW CONTINUED]

[0:36:47.8] JM: How does fitness factor into your life as a programmer?

[0:36:54.3] JS: At the very simplest level, if we talk about long-term benefits, if you die young from heart disease or you have a low quality of life, you don't have much time to actually benefit from the long-term benefits. If you're not going to care about fitness at all then you should probably try to make money real fast and spend it real fast because you're not going to benefit from the long term.

At a bigger level, I would say that there is — Okay. We can go really deep into what is the basis of human psychology. What are the basic human needs? What is it that has caused us as species to do things? Why do we have jobs? Why do we make money? I know this is going to be a little bit out there, but if you really, really think about it, for the most driving biological factor, which evolution would say, most scientist would degree with even though it sounds weird when I say in this context, is the sex urge, is to actually procreate.

At a very like basic level, I could say, "Hey, one reason why you should get fit is just for vanity's sake because it will improve your odds of procreating successfully," which is really important to a lot — This is funny, but —

[0:38:20.7] JM: Very reasonable.

[0:38:21.6] JS: Yeah. When I started the YouTube channel I started getting a lot of YouTubers, people that watching YouTube are 18 to 25-year-old single, kind of nerdy guys. What became very, very apparent to me was that one of their main concerns in life, which makes sense, would be, “Hey, how do I interact with the opposite sex? How do I get girls?” They don’t know how to do this stuff. I think fitness definitely plays into that. Confidence is more important, but fitness plays into that.

So just at the very basic level I’d say, getting in shape is going to make you feel better about yourself, you could feel more confident, make you have more appeal, which is highly beneficial. The other thing I’d say just from a more pragmatic standpoint for everyone that says, “Oh, I don’t really care about that.” You do, but, okay, you can talk to your own psychologist about that and dig deep into that.

Let’s just, from a very practical standpoint, if you’re healthy and you’re able to have high energy levels and you don’t have things that are weighing you down and you feel confident because you look good, you’re going to be more effective. You’re going to be more effective at what you do. You’re going to have more energy. It’s going to be easier for you to endure things for a longer period of time, and — Again, just to take one more trip back in the vanity zone. Guess what? Do you think that fit and good-looking people get promoted more or get more opportunities? I mean, what —

[0:39:50.1] JM: Yeah, of course they do.

[0:39:51.3] JS: Right. What world are we living in? We’re not living in a meritocracy. We’d love to be, but we’re not. I’m not saying that we shouldn’t have one, but we don’t, and it is probably not going happen anytime soon, so use every asset that you have, every advantage that you have, and that’s a huge one and it’s one that you can absolutely control.

I think it plays into a lot of — I always think of it this way, because a lot of people ask this question, “Why fitness? Why all these things? Why do you teach developers about finance?” It’s

like you got a car and if any one of those wheels on the cars is flat you may be overdrive for little while but pretty soon that rim is going to be grinding and bent and the thing is going to flip or the car is going to stop. A lot of developers, a lot of people in life, they got an awesome engine. They're focusing on — They've got these awesome wheels on the tire, but they're neglecting like one or two of them and eventually the wheels come off. Eventually the whole thing falls apart, and so you've got to focus on all these areas of life. You could be a super, super successful person and make a ton of money and if you don't have any kind of social skills, it's not going to be too much of a benefit to you. You could have a ton of money and if you don't have fulfillment in your life and don't know how to actually derive your own sense of fulfillment from what you do and have meaning and purpose in your life, it's not going to matter to you. You could have all the money in the world and you can try to spend as much money on doctors as you possibly can, but if you're overweight and you're out of shape and you've messed up your heart and you've got diabetes because of what you've done to yourself, what are you going to do? How are you going to spend that money? It's not going to be valuable to you.

I think it's important to realize that it's all connected, and as you bring up one area of your life, as you become more fit and you develop the discipline and all those things that are required to become more fit, you're going to find that that carries over to your career. You're going to find it carries over to your relationships. You're going to find it carries over to all these other areas of your life. That's why I go down that road.

[0:42:00.6] JM: Yeah, I agree with those things. I really like the idea that it's like your Simple Programmer is really about this holistic, like viewing the life of a programmer from all these different angles because they are all intermingled and to focus on just one aspect of those things would be at the expense of all the others because they all are intermingled and they all compound against each other. You write about this. You do videos about this. You podcast about this, and there's a lot of authenticity. I think authenticity is something that you willfully want to bring to the table and you do reflect that on a lot of your writing. Are there things that we can't talk about, like things that we can't say or things that you can't say, like controversial issues?

Because I know there are certain issues that — I try to talk about a lot of different issues of engineering. There's an things are so hot button or so just like you can't talk about. I don't know if you've read this essay by Paul Graham, it's called What You Can't Say, and his wife, Jessica

Livingston also wrote about this. They founded Y Combinator and they just talk about some of the most valuable pieces of advice, about startups at least, are things that you cannot talk about. I think there are certain taboos. Obviously, this is kind of a meta-question because I wouldn't expect you to actually say these things. Maybe, are there certain areas that they're so taboo, they're so controversial that you would at least suggest people examine, like more fundamentally because nobody can actually write about them? Even the that John Sonmez's of the world can't tell you authentically how to think about thing X.

[0:43:50.8] JS: I would say, honestly, I know this will sound fairly bold and maybe someone will try and call me on my bullshit here, but for John Sonmez, for what I will talk about, there is no X. There is no nothing. I mean I have done videos on porn and masturbation, because there's no one talking about this. I have to address this to my audience, because my audience is facing these problems and I have to give you an objective, a non-moralistic viewpoint on this, and I've done it. I've talked about my own experiences with it and why it affects you negatively and how to overcome these things because no one else is talking about this stuff.

Here's the other thing, is not everyone should be and not everyone could be. I agree with you in the sense that there are things that can't be talked about. Okay. I've got nothing to lose for a couple reasons. One; because I'm independently wealthy from my investments. I don't have to work. You could take away — I could lose my sponsor — I don't give a shit. It doesn't matter to me. Second; because I made a decision in my life that I'm going to live my life as fully as possible and whatever consequences come, they come and that's part of life and I'm going to deal with them regardless of what that means. That may mean that is going to cut my life short. It doesn't matter. That's a decision I made a while back just because of the way that I was living my life in such a guarded and protected fashion. I'm willing to go there.

At the same time, not everyone should be. I did a couple videos on this because people were asking me, they're like, "Oh, should I talk about politics or religion?" I'm like, "No." Let me tell you why. If you are creating a blog and a podcast and you're about — Let's say that you're specializing in AngularJS, JavaScript. That doesn't have anything to do with that. You need to be controversial. I talked about being polarizing, and I think being polarizing is extremely valuable, but you have to be polarizing along the right holes. For me, that's everything.

Okay, I won't say everything, because it's everything that will give you some kind of wisdom or value in your life. I'm not going to talk about politics for the sake of talking about politics or just because it's my opinion. I'm not going to talk about stuff that I do not think will bring actual value to people just to be a controversial figure. I have no interest in being a controversial figure for the sake of being controversy, but like I said, I will talk about porn and masturbation and do video on that because that is relevant and it is going to upset people, that's fine. I've done some videos on racism. I've done some videos on white male privilege, and I've gotten a lot of heat and flat for that, but the reason why did those videos was because I wanted to stop people from the victim mindset and mentality and that was — It wasn't to get views. It wasn't to raise a controversy.

Again, if you go back to that JavaScript developer that sees my videos and sees my content and says, "Oh, I want to be like John. I'm going to do a video on Trump or on transgenderism," or whatever they're going to do a video on. I'm like, "No. No. No. No. You're missing it. Be polarizing about JavaScript. Say why everyone should forget about Angular 2 and stick to Angular 1, and this is why I think that or why we should just throw out X-technology completely, or why you only develop on a web IDE, or whatever it is. Pick something controversial that's polarizing that's specific to your things.

Again, there might be other people out there like me that are teaching personal development for software developers, and in that case, sure, it probably is going to make sense for you to expand the boundary a little bit, but also be wary of your job and what your — Again, like I said, I come from a place — I told you the two reasons why I'm willing to go wherever I need to go is because, one; I don't have any — There's no boss of me that can fire me or not like my post on Facebook and give me repercussions. I'm my own boss and I don't even need the business. I've got real estate investments and money setup that I'm set for life. Not to brag, but I did that purposely because freedom is my highest value in life. Secondly, the fact that I'm willing to live by a sword, die by a sword. That's a philosophy that I adopted. Not everyone is willing to make that choice.

I'd say there's definitely some accuracy, a lot of accuracy, and what you said is that there're lots of topics that it might seem a little hypocritical sometimes when I say it, but I tell developers all

the time, “Don't just be controversy for controversy. I'm not saying share all your opinions and your politics and religion and don't hold anything back. I'm just saying —

[0:48:46.8] JM: Pick your battles.

[0:48:47.4] JS: Yeah, exactly, and make sure it's relevant to your personal brand because we all have a personal brand and as long as it's relevant. For me, like I said, it's a pretty wide thing, but I feel like someone's got to do it and that's a role that I've taken. For most developers that are listening to this advice, it's not going to be in your best interest to broach most of these subjects. Go crazy on the controversial stuff that has to do with the technology. Don't be afraid. Don't let you people shunning you, hold you back from talking about those things. As far as those other bigger political issues and stuff, it's really not going to help your brand in most cases.

[0:49:29.5] JM: Okay, now that we've talked about the substance of the content that you make, as a content creator myself, I'm curious to hear your thoughts on the different mediums, because there are so many different mediums that you can go after these days. You can write. You can podcast. You can do video creation. I'm just asking for myself, because I'm trying to build this as a larger media channel. How do you see writing and podcasting and video creation? What the differences across the different categories? What have your learning's been?

[0:50:05.8] JS: Yeah, that's a good one. That's interesting. I've seen people sit successful on each of the mediums and what I have found is that what seems to work best for me — I definitely got some favorites here, but a general philosophy in policy I think is to pick one medium to start off and be as prolific as hell on that medium.

One of the main strategies that I have in life that almost is always successful is just to be as prolific as help, because when you're super prolific, that's getting up to bat and swinging a whole bunch of times. You're going to hit — Not only are you going to get hits eventually, but your arms are going to get so strong from swinging those bats that you're actually going to get that bat. You're actually going to get good, and when you do get hits, they're going to be good hits. That's one strategy think that you need to make sure that you employ.

I think a lot of people, they take a scattershot approach. They're like, "Okay, I'm going to create a podcast a blog and a YouTube channel and social media and all that," and all that is great and it can be really good, but the problem is if you're only putting a trickle on each one of those, it's not going to be as impactful. That's a thing.

I would say focus on one, and as far as which one to focus on, I think there's a couple of things that come into play here. In my opinion and my experience, because I've been on all those mediums, video is the best format because of a couple of things. Okay, with video I've had podcast and blog for a long time. If I'm at a developer conference, yeah, sometimes someone will say, "Hey, I saw your porn site." "Oh, you're John Sonmez." They're like, "I heard your podcast."

Literally, being on YouTube, I'll be walking on the street. Just this last week, twice, when I went for a run, when I stop my run, people came out to me and said, "Oh my God! It's John Sonmez." Because you actually create — It's like being on TV. It's like being a celebrity.

That is powerful because not only people are hearing your words, they're seeing you. They feel like they know you. They feel like you're a star to them, and so that is very very powerful for building a brand. Just having that visual element.

Now, and I didn't expect this. This is not something that I expected because I have my blog for a long time and right now I get about triple the views on my YouTube channel as I do page views on the blog. The blog gets maybe like 250,000 views per month, but the YouTube channel gets 700,000 views on videos per month. I'm putting a lot more content there. I'm putting out two to three videos a day there.

What I would say though is — That's one factors, is I think video is the best medium possible, and then I would say that podcasting is probably the second. Again, because you're in someone's ear for a long period of time, and then and I think blogging is the worst right now. Although it's also has that lowest bared entry.

When I tell developers starting out, I say create a blog. I've got a free blogging course that I usually send them to and I say start there because everyone can do that. It's got a low bared entry. It's a good place where you can start writing and you can start — Writing is the key skill that's going develop any kind of communication for you anyway. You're not really going to be able to talk it front of a camera if you can't write for the most part, because you have to learn how to organize your thoughts.

I would say the other factor that influence all of these is where is your strength? Where can you — Or where do you want to go? You could develop strengths, but if you are really a strong writer and you don't have the camera presence and you don't want to develop the camera presence, then hey, then blogging is going to be it for you. If you got a great radio voice, but, again, you don't have the camera presence, than podcasting could be it for you.

Whatever you do, pick one medium, be prolific on that medium and then expand to the other ones, and that's how you're going to find the most success. If you look at people — If you just look at social media, you look at Instagram, you look at blogging, you look at Facebook and YouTube, you will find that most people that have a very large following and a very large brand have one place that they are super prolific on. That's their main thing, and I think that's important to recognize because there — Even if you look at someone like Tony Robbins. I think I went over to his YouTube channel and I think he had maybe a few hundred thousand subscribers. Not anywhere — You would think he might have millions of subscribers, but he doesn't, or Tim Ferris. I think the same deal there. His podcast is huge, but his YouTube channel, not so much.

There's other people like PewDiePie, he's got a ridiculously huge YouTube channel, but I don't think he has a podcast and he probably has a blog somewhere but I've never seen it. Those are the kind of things I would keep in mind.

[0:55:05.9] JM: You wrote a book about soft skills and I've heard several podcasts where you talked about that. Your more recent book that you're working on is *The Complete Software Developers Career Guide*. What's the diff between those two books? What is evolving in the narrative that you're trying to tell?

[0:55:27.8] JS: Sure. With soft skills, *Soft Skills: The Software Developer's Life Manual*, it was all about more on the person of element. There was a section on career. There was a section on fitness, in finance, in productivity, and learning, and marketing yourself, and then on more of the mental game, the self-development. What I called spirit in the book. It's was more on the holistic approach that I generally teach on my channel and on my blog.

That was important for what I feel like is really important just for everyone to grasp in life, but there was a lot of people that said, "You know, that career section — Everything was great, but that career section, I wish you would've expand on that. There's so many questions I have on the career."

I looked up there and I was like, "Okay, there's really not a really good soft skills book on just the career of a software developer telling someone how to get into the industry? What they need to learn? How to learn properly? How to do things, like deal with your boss and deal with coworkers? How to deal with prejudice, perhaps in the office? How to dress? How to go to an interview? How to get a raise? How to set yourself up for a promotion? All those type of things, what exactly do you need to know?"

I said, "Okay, two things were going on in my head. One of them was, "Look, I'm moving more into this personal development space, but I had a lot of success in my career as a developer and I've talked to, and coached counseled so many developers. I've got so much wealth of information from my audience and from all these interactions I've had. I want to capture all that." That was going on.

Then the second thing was I really want to create the ultimate, the complete guide for software developers where every single developer should have this book on their shelf. It's going to be valuable for everyone and it really is what someone would see say is the complete career guide that's all soft skills approach. Not technical information, but all from a soft skills. This is the tome, the manual you need to have a successful career as a software developer and answers all of your questions.

That was the driving force. I'd say a distant third would be just my masochism of wanting of the pain of writing another book. That's where it came about and that's really the difference is that

this book is completely focused on your career as a software developer from start to finish. It goes from the beginning of when you're starting out in your career and just getting into the field of software development all those questions; should you go to college, or boot camp, or self-educate. In whatever choice you choose, how is the best way to do it? To getting your first job or getting a new job as a software developer and then transitions into all the stuff you need to know. There so many things that new developers and even existing develops don't know that they need to know, like's source control, the basics of that, QA, continues integration, all those type of things. What is web development? What's embedded systems? What are all of these things at a high level?

Finally, going into the last two sections which are, as a working software developer, what do you need to know? This is where I talked about the dress and working with your coworkers and boss and all of these. , the last section is more for, “Okay, now if you to advance your career, where do you go from here? How do you actually to take it to the next level? How do you do something like consider creating a personal brand or starting podcast or getting out there and speaking at conferences? I try to pretty much — It's an 800 page book at this point, so I try to pretty much encompass everything.

[0:59:15.3] JM: A lot of the things that we cover on Software Engineering Daily are about what's changing in the technical world. For example, platform as a service makes it a lot easier to build and deploy apps that are reliable and scalable. You've got things like Firebase that can handle your database more easily and it's all low-cost. It also changes like it's becoming easier to hire contract developers, like contracting got a bad name in the 90s and the early 2000's because you couldn't do quality control but now we've got really good talent marketplaces. It's much easier to hire contractors.

There's a lot of systemic changes to the world of software. If you're thinking about building a software product, it's gotten easier in certain ways. Does that impact how you think about the career guide for a software developer?

[1:00:12.5] JS: In some ways, yes, but I always try to think in terms of principles rather than in what I would call strategies or tactics.

[1:00:24.1] **JM:** More timeless stuff.

[1:00:26.1] **JS:** Exactly. Yeah. From everything I try to do, I try as much as possible, because evergreen. That's, like you said, the timeless stuff.

From the books perspective yes, there are probably some things that will change. Some of the tools that I called out the names might change. I'd say 95% of the content is going to be more the higher principal level stuff. That's probably not to change, or like if it does, it'll be adaptable. You'll be able to figure to figure out, "Okay. Yeah, yeah it was this, but this is what it means now. It should be pretty obvious from that."

Not that I won't update the book ever, I probably will update the book but I've always tried to focus on those, those higher level things. When I first was approached to write my first book, *Soft Skills*, Manning had approached me and they said, "Hey, you've got a pretty big audience. We'd love to — Whatever book you want to write, we want you to write a book. How about android development? Because we know you've done some course —" I was like, "You know what? If I'm going to spend a time to write a book, it's going to be a timeless book. It's going to be one that's going to sell for the next 10 years. It's not to be some technology." They're going to shock, they're like, "Well, you did all these Pluralsight courses on technologies and stuff. Why wouldn't you want to — You could easily catapult off of the back of at that." I said, "No. No. I'm not going to write a book to make money. It doesn't make sense. If I do, it's going to be something timeless."

I've tried to do that, like I said, with most of the material that I've put out, but definitely with books because books, you don't make much money on. It's true, and so you want them to sell for a long time. Really, books should be something like that. The most valuable books I have on my bookshelf are books that are timeless, that have principles in them, that are not tactical-based. There's definitely some value and tactical base things, but they changed and they got a date. Yeah, the field with software development, it's going to change, the way that we write code is going to look different, but the higher-level principle is not going to change. Algorithms will always be algorithms. The process you go through to solve algorithms, and a lot of the ways that you deal with your careers and a lot of the ways that you deal with people, because you could almost say that everything that I write about is how to deal with people because that's

ultimately what all of us software developers are not do more than anything else and that's ultimately what defines our success. Those things won't change.

[1:02:52.7] JM: Building a software company is based in software that you're selling that feels like a very different business than building a media company. Does it ever feel like — I'm speaking firsthand. Software Engineering Daily is a treadmill where I have to produce content on a regular basis, and certainly there are some evergreen qualities to it but it's not exactly the same as writing Windows 95 and just like selling that in perpetuity and all you have to do to make it keep going is just printing more copies of Windows 95. Does it ever feel like the media business that you're in conflicts with the idea of software engineering, which is like building automated scalable systems?

[1:03:41.5] JS: It's kind of interesting, because that you would say that because there's a real big fallacy that most software developers have when they want to become entrepreneurs, which is they think that there is such a thing as passive income, and there's not. The closest thing to passive income is real estate, to be honest with you. I have real estate investors and say, "I hardly have to do anything with them. It just earns me money every month," but it costs a lot of money to do that in time to sit that up.

With software — This is why I say this is one of the biggest fallacies, because I get these emails all the time. All these people send me emails with their million-dollar ideas and I have to crush their dreams and I have to reply back and say, "You know what? While I was sitting on the toilet this morning, I had \$10 million ideas and they were all just as good as yours." I'm not saying it to be mean. I'm not saying it to be conceded, because I believe everyone has million-dollar ideas. It's execution, but is not just executions. See, that's the other thing, is software developers think, "Oh, well. I'm in a unique position. I can build this software. I can build this awesome software."

You know what? The biggest piece that you're missing is marketing, because that's the one thing is like marketing, marketing, marketing. It doesn't matter. Let's say you build your awesome piece of software. If you don't have a way to sell that, if you don't have a sales pipeline, a sales funnel, a marketing funnel to be able to get people into that funnel to actually get exposure to what you're selling and then buy it, it doesn't matter. You could have the most

fantastic piece of software ever built but if you don't have that, you got nothing. You just wasted a whole bunch of time.

A lot of developers, they build these awesome applications and they think that they're just going to make money. They think if they build it they will come, and it's so so untrue. They think that it could be automated, that if they just get this running — Yeah, I'm not saying that there's no way to automate stuff. Actually, for a good a good portion of my business is automated, right? I have a bunch of courses that I sell and I have a bunch of YouTube videos and podcasts and blog posts and I capture email addresses. People come to my blog and they sign up for a course, a free course and they get put their email address in or they find a YouTube video and they sign up and they automatically go through an automated email funnel that will send them emails that will eventually offer them and pitch them my courses and they sell automatically, but that won't run forever. I still need to create more content and I also need to adapt to the marketplace. I could take my hands off the wheel and probably for the next five years I think the revenue could probably still be maybe like 20 to 30k a month just with no one touching it, but it'll eventually go down and it could eventually drop off completely, and Google could shut me down. There's a of factors that are involved in that. If I want to keep the business going, I'll have to do it.

There's always a treadmill to business. I would highly advise to make things as automatable as possible, but I'll point you to a really good book though for the treadmill kind of thing. It's actually a really strange book, another one by David Data called *The Way of the Superior Man*, and the very first chapter or the book is this what I'd like to highlight. The written version is important because the audible version is a totally different thing, so the Kindle or written version.

In the very first chapter of the book he says something that highly impacted me which he talked about this idea of like, "It's never going to be done. It's never to be finished. Stop waiting for things to be completed. Stop waiting for things to be finished. Stop saying when this is done. Just realize that —" and I'm paraphrasing here, but essentially the battle begins anew each day.

I lived a lot of my life trying to say, "Okay, I just want to be done with this stuff. I just want to be off the treadmill." What I realized is that, "No. No. N. That is life," and as soon as just having that realization that no matter where you get to there's always going to be new problems. There's always going to be new — You don't ever — You don't go for a run and get your diet in order

and lift some weights and then you're done, like, "Okay. I'm fit now, or am I fit now?" No. Every day you got to wake up and you got to get gym again and you got to go eat healthy again and you got to go for your run again.

If your mindset is — This is why a lot of people fail on diets by the way, is because they think, "Okay, well I just got to do this six-week program and I got to get through till Friday and then I'm done." You're living in misery the whole freaking time. People live their business life that, they live their lives like that, they with their career like that, and the whole time they're waiting for their life to be done so it can begin, it's insanity, but we get trapped in that. My answer to that this is to say, "No. No. No. If you recognize that there is always going to be a battle. That it's always going to be harder. There's always going to be problems. If you're going to do a diet program, is a lifestyle. You're going to be doing this for the rest of your life, but that's okay and you embrace that."

Now, everything is a challenge. Now, your perspective is different. Now, you're like, "Okay. I'm to keep recording —" For me, I'm doing two to three YouTube videos a day. I'm not looking forward to the day when I can stop. I'm not saying I'm going to do it forever, but I'm saying that every day I wake and I'm like, "All right. Let's create some more videos today." That's the shift and the difference there that I think is more important than the issue of what kind of business is more automatable or not or can you ever have a passive income business, because the chances are if you're looking for that and the reason why you're looking for that is so that you can get off the treadmill of life, you're going to be sorely disappointed because it's never did happen. Instead, you just need to embrace the treadmill or life and say, "Look, there's always going to be — I'm always going to go to the gym and have to lift weights to maintain what I've got. I'm always going to have to face new a challenge or fight new battles or have a fight with my — I'm never going to have perfect relationship. I'm going to have fights with my spouse and my kids and all that, but that's cool. That's all right. That's okay. That's what life is. I'm not waiting for everything to settle down. I'm jumping into the fray."

[1:10:01.0] JM: Okay. Great way to conclude. John, thanks for coming on Software Engineering Daily. It's been great talking to you.

[1:10:06.8] JS: Yeah, thanks Jeff. You have brought a ton of depth, I have to say, the questions that you've asked. You've asked just fantastic. One of the best interviews I think that I've had the opportunity to be on.

[1:10:18.1] JM: I appreciate that. I appreciate that, and thanks for bringing all your content. I remember I listened to your Software Engineering Radio about soft skills. I was pleasantly surprised with how interesting it was. Cool. Good material.

[END OF INTERVIEW]

[1:10:36.0] JM: At Software Engineering Daily, we need to keep our metrics reliable. If a botnet started listening to all of our episodes and we had nothing to stop it, our statistics would be corrupted. We would have no way to know whether a listen came from a bot, or from a real user. That's why we use Encapsula to stop attackers and improve performance.

When a listener makes a request to play an episode of Software Engineering Daily, Encapsula checks that request before it reaches our servers and filters bot traffic preventing it from ever reaching us. Botnets and DDoS are not just a threat to podcasts. They can impact your application too. Encapsula can protect your API servers and your microservices from responding to unwanted requests.

To try Encapsula for yourself, go to encapsula.com/sedaily and get a month of Encapsula for free. Encapsula's API gives you control over the security and performance of your application. Whether you have a complex microservices architecture, or a WordPress site, like Software Engineering Daily.

Encapsula has a global network of over 30 data centers that optimize routing and cache content. The same network of data centers that is filtering your content for attackers is operating as a CDN and speeding up your application.

To try Encapsula today, go to encapsula.com/sedaily and check it out. Thanks again Encapsula.

[END]