EPISODE 1300

[INTRODUCTION]

[00:00:01] JMeyerson: Software is getting a lot better at all areas of the stack. And what does that mean? What it translates to, in many cases, is that software simply becomes higher level. Take GitLab, for example. So what is GitLab? GetLab is basically an open source, high-level loose collection of DevOps tools that gives you an introductory experience that is not bad out of the box. So that's what GitLab does. Why do I mention GitLab? This show is not about GitLab. This show is about Better Stack. What is Better Stack? Better Stack is a company that was referred to me by Leo Polovets of Susa Ventures, who sends me the most interesting companies to look at. Seriously, he's probably sent me – Over the last five years, he's probably sent me, I don't know, 30 companies that he's invested in. And those include, I think, if I get this right, Mux. He send me Mux when they did the seed round, which is just very characteristic of Leo. Just smart enough to see Mux far before anybody else.

Anyway, this is not an episode about Leo. I hope Leo comes on again in the future. This is an episode about Better Stack. What is Better Stack? Better Stack is one of the coolest infrastructure companies I've seen in a while. Why is it so cool? Better Stack does something new. And what that new thing is, is that they're like GitLab for monitoring. And I think that's my tortured analogy. I don't think that's his, Juraj. He's the guest for today's show, Juraj Masar. So he runs Better Stack. And Better Stack is essentially like this unified hub for DevOps tools that he creates. It's like a full stack, him and his team. It's a full stack monitoring solution. It's out of the box tracing, and I think metrics, and maybe logging. And the essential idea is like we're giving you an out of the box experience for metrics, and you're going to like it. And maybe it's not as good as Datadog in some ways, but it's actually different than Datadog. Datadog is a very deep product suite at this point. And what we know about monitoring is that it is one of these fields that just reinvents itself all the time. The reason it reinvents itself is because monitoring, almost like the smartphone just needs to be refreshed every few years, where you basically say, "Okay, let's start from zero and ask ourselves, "What can we do in machine learning? What can we do in user interfaces? What is React do these days? What is GraphQL do these days? We have all these things together, what do we get? Do we get anything special?" We do get something special today.

So the TikTok of yesterday's metrics platform to today's metrics platform has gone through a period, and at the end of that period is Better Stack. Better Stack is a very new type of product. It is a fully integrated GitLab-like experience for monitoring. Juraj is a fantastic guest. I am very excited about this company. I think you're going to like this episode if you're at all interested in cloud operations, and DevOps and that kind of stuff. Thank you.

[INTERVIEW]

[00:03:09] JMeyerson: Jay, welcome to the show.

[00:03:11] JMasar: Thank you for having me.

[00:03:13] JMeyerson: I don't usually start out shows this way, but I found your company through a venture capitalist, who I'm friends with, named Leo Polovets, who's been on the show. And he has great taste in software companies. I think Leo is an example of somebody who is continually able to find deals that other people are not seeing and find a lot of value. So I'd like to start by just asking you, to the extent that you understand him and know his portfolio, how does he select software companies to invest in?

[00:03:46] JMasar: The funny part is that we actually did the deal with his partner, Chad, right? So I read Leo's essays before, obviously, before Susa Ventures invested. But it was a funny coincidence. So frankly, you need to ask Leo about his magic. But I do agree with you, it's not a coincidence. There is a pattern in the choices they make.

[00:04:08] JMeyerson: So when I look at your landing page, the first thing that comes to mind is, and forgive me if this is wrong, but the first thing that comes to mind is the GitLab page. So I don't know if you've ever been to GitLab, but GitLab has all these bells and whistles. When you go to the page and they say, “Okay, we basically will replace every piece of software in your entire software development lifecycle workflow. Here's how we're going to replace it.” And I look at your website, it kind of looks like that's what you're trying to do in the monitoring domains. You're basically saying there's so much monitoring software out there. We're going to build a unified system for it. Is that an accurate way of understanding what you're up to?

[00:04:54] JMasar: Partly. Well, let me ask the question by asking a different question. Does it make sense for any company, any tech startup running a website or a web app to have one app for monitoring purposes, one app that actually monitors an API? Another app that calls the right person on the team, wakes them up in the middle of the night to fix the issue? And a third app to show a status page to your clients, right? So essentially, what I'm talking about is, say, Pingdom as the first company, PagerDuty as the second company, and Statuspage.io from Atlassian as the third company. It doesn't make sense for most of the tech startups out there to use these free tools in conjunction. Where if you think about it really, what is incident management? Is essentially passing the information that's something's wrong, and then do you need to fix it within a company, right? To the right person, anything.

What is a status page? Well, it's the same thing really, which is doing it externally. And when you connect these tools via integrations and API's, some things get lost in translation, right. And so if this was an MBA, we would say that most businesses can be described as bundling and unbundling of what came before. So this is one way of looking at it. From my perspective, I'm fixing my own itch, right? Like I'm solving my own problems. And monitoring is as old as the Internet is, but some issues were not solved before we started digging into this.

Let me give you an example, your site goes down for two minutes, but it takes you five minutes to open up your notebook. And when you refresh your product, everything seems to be working great. So what is the number one thing going into your mind? Well, was this a fake incident? Or was it a really short incident, right? And all the monitoring tools that I used before would just tell me, "Your API was giving me 503, and right now it works." But what we do is that we take a screenshot of the error page, we save the error message, and we show it to you in the very first incident report that you get, right? Like your site went down for two minutes, and it says that server cannot connect to a database, right? And this is really not a rocket science. This is something that the old monitoring tools should have as a feature, should have had as a feature for ages. But for some reason, they didn't, right? And I cannot really answer why they didn't have it before, because it makes so much sense for me. So, frankly, we just build it something that we as developers wanted to use ourselves.

[00:07:26] JMeyerson: The conceit makes sense. You would want all of these things in one system. Or I put it another way. I think, for companies that have been around for 10 years, maybe it makes more sense for them to use the disconnected systems that they've been using for a long time, because it's so tightly knit into their infrastructure. But if you're building a new company today, you want to be operating at a higher level. And this is something that bundles several applications that have synergies with each other together and lets you work from a higher plane.

[00:08:07] JMasar: Yeah, you sold it very well.

[00:08:11] JMeyerson: So why is that? Why are we now in a time where we're bundling this stuff together? And why has it been disconnected in the past?

[00:08:21] JMasar: Well, if you look at companies like PagerDuty, which is a YC company, and the history of the company is very well documented. For some reason – Well, Jeff, help me out here. How old is incident management?

[00:08:34] JMeyerson: No. It's as old as software itself. I mean it's just the incident –

[00:08:38] JMasar: It’s solved internally at the companies, right? Like when was PagerDuty founded?

[00:08:45] JMeyerson: 2006, 2007, or 2009?

[00:08:50] JMasar: That’s actually what I'm getting at, is that I think these companies started as incremental changes to the status quo before. And right now I just feel like we're ready for the next structure, right? And I'm not being romantic here. We're just sort of developing software that makes sense. So that it just makes sense for it to exist, if you know what I mean. It really is not rocket science. The number one thing people tell us is that it's super simple to set up the software. I don't want to sound like a sales guy here. So let me put it from a different perspective. When I've been in different VP of engineering, CTO roles for ages, right? And I've tried all these infrastructure monitoring tools. And let's talk about uptime monitoring specifically for a second. I use a number of international phone numbers, right? And one of them very naive thing when you have a number, anonymous number, when you sign up for any monitoring tool, what you’re thinking is that is this tool actually going to call me? Does it call international numbers, right? How is the alert when it actually goes off actually going to look like, right? And for some reason, it is extremely hard or laborious to get a test alert from most of these services, right?

So in a very abstract way, what better uptime is selling you is a better service, but mainly a better user experience that you're used to when you're using b2c apps. Like say, well, anything that you have on your phone really. But the strength of making the software easier to grasp, easier to use, it's finally getting to enterprisey tools that you use for work. And I don't want to talk shit about competition or talk shit about anybody else, but finding one person who is going to tell you that, “Well, I'm an SRE. And, man, I love the PagerDuty dashboard. Man, I love the AWS dashboard. It’s just so intuitive, and it makes sense.” Like nobody's going to ever tell you that, right? Essentially, that is the entire business model behind DigitalOcean. DigitalOcean can only exist because it does less than AWS, right? And I think that this is an overall trend that you are going to see in most of the b2b infrastructure SaaS tools as well.

[00:11:12] JMeyerson: All right. So here's my thesis. We're at a point where you have new kinds of web-based companies that basically position themselves like domain-specific operating systems in the cloud. And the reason for that is we've solved the frontend, because we've got React. We've solved the backend, because we've got really good cloud infrastructure and Kubernetes. So now, it's a question of what does software look like when you've solved the interface and you've solved the backend? It looks a lot better. It looks a lot easier to use. So if you look at GitLab, GitLab is like, “Okay, we're the operating system for your DevOps.” If you look at ClickUp, they're like, “We’re the operating system for your productivity tools.” And then you look at what you're trying to do. And you're basically kind of having an operating system level approach of monitoring and uptime. You're saying this domain is so big, that we need to basically like make an interface and like a system for managing these different domains that are the sub-domains of monitoring.

[00:12:24] JMasar: Essentially. But in a nutshell, I'm trying to delight our customers. That's what I'm trying to use. And how come any monitoring software can have a high NPS, Net Promoter Score? Frankly, do people actually like using infrastructure monitoring tools, because whenever you need to open them up, something's wrong, and you need to fix it, and you're having a bad experience anyway, right? So counter intuitively, I don't want our users to really think about better uptime, or using too much. I want them to use it as little as they need to. At the very beginning, I want to delight them, right? So even if you have two pieces, two almost identical products feature-wise. At the end of the day, I feel like people are going to gravitate towards the product. Give them a better user experience. And that is essentially I think what you're trying to say. We solve the frontend. We solve the backend. We are able to ship software faster than ever. So finally, we can get to the ultimate problem, which is creating delightful products. And at the end of the day, your customer doesn't really care if your backend is built in Java, Scala, Next.js, whatever, right? They don't care. They care, “Is the app fast? Does it do what I want it to do? And how painful is it for me to work with the app?” And if some something breaks, is there somebody to ask for help, right? That’s the thing.

So, ironically, most of the time, I'm thinking about user experience and product. And software engineering for me is mostly the canvas that small startups use to deliver that amazing user experience. Does it make sense at all, Jeff?

[00:14:03] JMeyerson: Yeah, totally. Okay, so we've got kind of the vision laid out for people. How does that lead to the product strategy? Like where did you start with the product? What was the first set of things you built? How did you – I mean, I understand where you're going conceptually with the company. What was your entry point for the for the product? What was the MVP basically?

[00:14:27] JMasar: Sure. It's extremely funny. It’s still online. You can go to betteruptime.com/homepage-v1, and you can see the very first homepage. By the way, I highly recommend everyone doing this. Whenever you do a major redesign, like save – And this not necessarily design-wise, but when you shift the way you talk about products, save the previous version so that you can contrast and copy, because time flies by in startups super-fast.

[00:14:55] JMeyerson: Betteruptime.com/homepage-view?

[00:15:00] JMasar: Homepage-v1, as in version 1.

[00:15:03] JMeyerson: V1. Got it. Got it. Got it.

[00:15:05] JMasar: Yeah. And if you open it, it says call me when your website goes down, right? It says cannot afford the downtime. Get alerted with the fastest optometric service. Free plan included. As you can see, we're still searching for what is the main value prop here. At that point, we're advertising this is the fastest uptime monitoring service. It turns out people don't care about that. They care about reliability. But there is one amazing aspect about this landing page that I love, is that this is the entire onboarding. You can see free fields on the page. What's your work email? What is the URL to monitor? What is the phone number to call? CTA, start monitoring for free. You click it and you're like, “Okay, do you want to get phone calls as well? Our all you can eat plan includes all the international phone calls. Enter your credit card here.” You enter a credit card. Done. Boom! This was the MVP. It calls the right person on your team. No BS. Like one page to set it up.

And if you think about all the other monitoring tools, like give me one example of a service which is so easy to set up, right? And this comes down to I'm not the kind of guy that enjoys tweaking my Prometheus config, right? What I like is actually running my business and building amazing software shipping yet another feature, right? And I have my Prometheus in place so that the servers don't crash, right? But that's it. It's an insurance. Essentially, better uptime is an insurance policy that you are purchasing so that you know that your software works.

[00:16:38] JMeyerson: Okay. Again, that's great. Pretty high-level. Tell me a little bit, like at a deeper level. So like are you installing an agent? How is this thing working?

[00:16:51] JMasar: Right. So Better Uptime, essentially, is an external monitoring. So you enter the URL and your contact info, and you're done. We are monitoring your website from multiple regions, depending on your configuration from multiple locations. And when we verify that there is an incident going on that your site is down for some reason, or your CDN is down, or whatnot, we notify you in the best way possible for you. It's as simple as that.

[00:17:22] JMeyerson: So how much information can you get about a company by that kind of external monitoring?

[00:17:30] JMasar: Well, anything which is publicly available, or privately if you first did use some private credentials for monitoring, private API's. But anything that the browser can see, headers, timing. But frankly, the most interesting thing is not about one particular request or response header that we get from your server. It's about the time series of different events that led to a particular incident, right?

[00:18:01] JMeyerson: So let's say we get to the point where I have an incident. Like I'm woken up at three in the morning. I have no idea what's going on. What is Better Uptime telling me?

[00:18:11] JMasar: It tells you the these were the three error messages that your web server shown as it went down. So it started with a long request that timed out. Then the first error message was that your web server cannot connect to a database. And then whatever, your Redis ended up failing as a result of being overloaded, since other things didn't really work out. So it shows you the time series of the public events. At the same time, this is the moment typically where you would shift to our other product. So the company that I run is called betterstack.com, right? And it is a collection of tools. And Better Uptime is the very first product people find and use. This is what we've been talking about so far. It is uptime monitoring with incident management built-in at a status page. If you're investigating an incident, you do need to see metrics, logs, maybe traces. You need to see the inner diagnostic data from your app. And at this point, you will typically switch to LogTail, which shows you exactly what's going on in your cluster.

[00:19:20] JMeyerson: Got it. Okay. So if I look at betterstack.com, I see two products, I see Better Uptime and then I see LogTail. So, basically, with Better Uptime, you're trying to have monitoring tools under a high-level application, like good UX world. And then LogTail, it looks like you're kind of trying to apply the same product design philosophy to the log management space.

[00:19:53] JMasar: That's correct. Yes. And we are blurring the gap between logs and metrics, because wood LogTain does is that it stores your logs in a structured form, right? So, typically, most of the Elastic stack based solutions out there, essentially mostly allow you to search your logs. But that’s it, right? What we do is that we automatically recognize commonly used log patterns for your logs coming from nginx and Apache, and generate dashboards, dynamic dashboards and give you generated dashboards based on this time series data in addition to letting you search the logs as well, right? So we are essentially blurring the lines between logs and metrics, if it makes sense at all.

[00:20:38] JMeyerson: Yeah. I mean, conceptually, I really like what you're doing with your company and how you've architected it. But convince me that if I go with LogTail, I'm going to have anything close to what I get out of Datadog. Full disclosure, Datadog is our biggest sponsor. But why would I use LogTail instead of datadog?

[00:21:02] JMasar: Right. By the way, Datadog is amazing. And the history, the company, I have a great respect for the founder and for the way the company was built. Datadog is super strong when it comes to metrics. LogTail is super strong when it comes to logs. And New Relic is super strong when it comes to APM, right.

[00:21:27] JMeyerson: Sorry. Wait. What is APM? What is a good APM?

[00:21:31] JMasar: New Relic.

[00:21:32] JMeyerson: New Relic. Okay. Got it. Got it. Got it.

[00:21:34] JMasar: Yeah, yeah. So as my favorite movie, I think it's called The Interview. The guy says it's same, same but different, but still same. It is solving the same problem from a different aspect. And with logs, with LogTain, we start with the highest alerting information with logs. By the way, one interesting company that I would think that you're going to ask me about would be Lightstep, because –

[00:21:59] JMeyerson: That's my next question.

[00:22:02] JMasar: I think Lightstep and Datadog actually are like extremely direct competitors. But at the same time, if you – Well, here's the thing. You do need to keep your logs. You do need to store your logs. And at some point, you do need to look into your logs, right? And with New Relic, Datadog, and Lightstep, logging feels like an afterthought. It's something that companies need to keep for compliance reasons. So we do metrics, we do APM. And you can also start a lot with us. When you log into LogTail, the number one thing that you see is logs, and you can dig into specifics of what individual workers and users are doing within your product. And metrics are the cherry on top.

[00:22:50] JMeyerson: Well, okay, so you mentioned Lightstep. So did you know Lighstep got acquired?

[00:22:55] JMasar: Yeah, I do. Yeah.

[00:22:57] JMeyerson: So my theory here is if Lightstep was actually in competition with Datadog, they would never have sold. So there must be something fundamentally different about the business.

[00:23:08] JMasar: It depends. Well, I'm not sure if the transaction details were disclosed, right? So I don't know to what extent you know in the background what's going on in the company. Multiple things that can lead to an acquisition, right?

[00:23:23] JMeyerson: Yeah, that's true. I guess. I mean –

[00:23:25] JMasar: Well, here's the thing. Again, Jeff, I think the companies and the products that you should also ask about is LogDNA and logs.io and —

[00:23:38] JMeyerson: Okay. Okay. All right. Al right. You've made your point. You've made your point.

[00:23:42] JMasar: This is an insane market, right? These people are willing to ship logs to paper trail for some reason, right? I'm just approaching the same people and telling them, "Okay, I do understand you have this fundamental needs to store logs. Well, here's the thing, here's a much better product at a significantly cheaper cost."

[00:24:03] JMeyerson: Got it. So back to the pie in the sky stuff. So betterstack.com – So Better Stack has two products, LogTail and Better Uptime. So okay, the other cool thing about building software these days is you can think at a pretty high-level and build the company at a pretty high-level and think about the long term at a pretty high-level. Are you thinking you want to eventually provide like a cloud experience? Like a hosting experience?

[00:24:33] JMasar: As in to compete with companies like DigitalOcean?

[00:24:36] JMeyerson: Yeah, or compete with – I mean, this is another one of these really big markets, like whether you're rendered.com, or Vercel, or Netlify, or whatever, you've got a great business ahead of you, right? So why not do a better PaaS?

[00:24:53] JMasar: Yeah, I'm searching for all these better X domains **[inaudible 00:24:58]**.

[00:25:00] JMeyerson: Well, what's the most random one you've gotten?

[00:25:03] JMasar: I'm not going to share that. But that's a great question too. Yeah, look, Internet is definitely growing. We're just getting started. So any solid Internet business definitely makes sense at least on paper. At the same time, whenever we start a product, I try to be at least 10X times better than the existing tools, right? If you compare Better Uptime to Pingdom, if you compare Better Uptime to UptimeRobot, it is definitely 10X better. If you compare a LogTail to the paper trail, it's definitely 10X better, right? And my question in my mind is if I weren't doing a DigitalOcean, what is the game plan there, right? And I think the game plan there is actually going for a cheaper cost, right? And by the way, while we are it, there is a company in Europe doing something super interesting. It's called Hetzner. Have you heard about it? It's called hetzner.com.

[00:25:59] JMeyerson: How do I spell that?

[00:26:01] JMasar: It's H-E-T-Z-N-E-R.com. And they are, I think, the biggest cloud hosting in Germany. So they do like 2 billion in revenue. They're not small by any means. And what they do is that – Actually, look up their certifications and look into their like visualizations and actually picture from their data center. They're budget, but with the rock solid infrastructure. Recently, there was this fire in the other server house. It was OVH, I think. When you look into that server house – I don't want to be too aggressive here. But it looked kind of cheap, right? Looking at the data center, I wouldn't want to host my info there. If you look at Hetzner, it is AWS level by any standard, right?

At the same time, look at the prices. Just go to hetzner.com. And by the way, I'm not affiliated with them. I'm just a delighted customer. Go to cloud, to go to prices, and look up the very cheapest instance. You get a better deal. You get a better machine that you do with DigitalOcean for like half the price. And it's actually a faster machine, right? The only tricky part there, which is actually the deal breaker for most listeners probably is that they currently have just two data centers, in Finland and in Germany. Well, more data centers, but two locations. They're only in these two countries. So, probably, if your user base is in the US, it should not be the first choice. But if you're ever thinking about what is the best AWS data center in Europe, well, maybe it's not AWS. Maybe it's Hetzner, right?

And you talking about another cloud company. Well, frankly, I would need to have a plan on how to be ` 10X better than Hetzner right now. And when they open their shop in the US, if I'm DigitalOcean, I would be kind of scared, because they know that their stuff. They're building their own servers. And they have a – I don't know, like 10, 20 years old history. So, frankly, I think in that space, the game is about the cost. And I think there are companies doing amazing job there. So it would not be the obvious route for me. The obvious route to answer your ultimate question about where I see this field going in like five to 10 years is integrating all the data. At the end of the day, the SRE doesn't really care if an insight is coming from uptime monitoring, logging, APM, metrics, whatever that is. They just want to find the root cause. And with Lightstep, what’s actually super interesting as a concept was to searching for that cause automatic with simple machine learning models, right? I think the gap between these individual product categories is going to be blurred going forward. And I think it's already obvious.

[00:29:08] JMeyerson: Do you know the company WP Engine?

[00:29:10] JMasar: Yeah, sure.

[00:29:12] JMeyerson: You've never used it as a customer, have you?

[00:29:14] JMasar: No. Is it Jason Cohen? Jason Cohen, right?

[00:29:18] JMeyerson: I don't know who the founder is.

[00:29:21] JMasar: Yeah. He, by the way, has amazing amazing blog, smartbear.com. He has amazing talks on business. I highly recommend them.

[00:29:30] JMeyerson: Did he start Smart Bear software?

[00:29:32] JMasar: Yeah, I think so. Yeah.

[00:29:34] JMeyerson: Wow. Okay.

[00:29:35] JMasar: Yeah, yeah, yeah, yeah, yeah. And by the way, he also has a great taste in uptime monitoring, I can tell you that. But yeah, yeah, yeah. So what about WP Engine? What did you want?

[00:29:45] JMeyerson: Well, okay. So I look at WP Engine, and whenever I'm going over my finances for the company that I – Software Engineering Daily. Like the softwareengineeringdaily.com runs on WP Engine, and every month I see the bill come in, and we're paying $400 dollars a month for probably a WordPress container. It's a WordPress.

[00:30:05] JMasar: It's amazing business, right?

[00:30:09] JMeyerson: 100%. 100%.

[00:30:10] JMasar: Because you're paying for your – You're paying for being calm when you're an airplane **[inaudible 00:30:15]**.

[00:30:17] JMeyerson: Right. So let me ask you a question. So what if we make the cloud hosting version, general cloud hosting version of WP Admin? What happens if you say, "Okay, you're going to have to pay $400 per container?" What kind of differentiated cloud service could you offer? If you basically say, "We're actually not the cheapest. We're the most expensive?"

[00:30:42] JMasar: That’s a funny one. Look, Jeff, I think I'm not inclined towards businesses that are solely about price. But the funny part is that, for instance, let's go back to Better Uptime. If you are replacing three different services, for instance, Pingdom, integrated with PagerDuty, integrated with StatusPage, that your total bill is going to be about 500 to 600 bucks, right? So actually, from my perspective, our pricing can be solid. We don't need to be the cheapest by any chance, because you're already paying a shitload of money for your existing tools, right? The bar is really high. It's super easy to be a little bit cheaper so that you as a product can be still expensive. You can still charge 300 bucks, for instance, for the same product, 300 bucks a month, right? But for the same thing, for the same number of services that you're wondering, you're still cheaper than the existing free tools that people are using. That is the beauty of bundling and unbundling. See how did the WP Engine?

[00:31:49] JMeyerson: No. No. I see what you did there. The point I was actually just trying to illustrate was just that there is, I think, that the category of hosting has a lot of depth to it. And I think it's like it's one of these perennial categories like what you've built so far with logging and monitoring. So, got it. So I understand what you're doing. I understand what you're doing kind of at a technical level. Where are you at right now? Like what are your biggest problems right now? What are you trying to do? What are your initiatives? What's your strategy? Or your tactics I guess I should say. What are the tactics?

[00:32:19] JMasar: Right. Shipping software faster, right? That is the number one thing. I, and the team over here, have a solid roadmap for the next 6, 8, 12 months, right? Like we know what we would need to build because there are companies coming to us and be like, “Can you do this? Can you do X?” And at some point you get better at pattern matching and product and know that, “Okay, this actually makes sense,” right?

So at the end of the day, the number one game for us is that how fast can we ship software of certain quality so that we actually be like the customers? That is the name of the game? How fast can we ship it so that we admit the quality bar?

[00:33:00] JMeyerson: I literally am publishing a book called *Move Fast*, that's about how Facebook build software, next month. And that's the whole reason I wrote the book is what you're describing. Like modern software company, really, the measurement of how well you're going to do is to some extent how fast are you moving? How fast are you morphing with the market? How fast are you improving your own software? How fast are you keeping up? How fast is your hiring process? How fast is your onboarding process? How fast is your operational expertise?

[00:33:36] JMasar: Do you know the underlying reason behind that? Because the faster you do all those things, the faster you'll learn. The faster we learn about the opportunity, about your customers. The faster you can be like, "Okay, screw this. This is not a good business. Let's do something else." Or this is a great problem to solve. We are just approaching it with a completely idiotic solution. We should do something else, right? So if you look at the flip side, if you're not moving fast, like do you think you're the visionary that when you come out with version one of your product, the software is going to be perfect and people are just going to throw cash at you? You just build it and they come?

[00:34:12] JMeyerson: Never happens. Never ever happens.

[00:34:14] JMasar: Never at all. I'm looking forward to the book, by the way. I saw the cover. I was like, "I'm going to order this. I read through it briefly before the podcast. I'm like, "Dammit! I cannot buy it yet."

[00:34:23] JMeyerson: You can't buy it yet. I'll send you a PDF.

[00:34:25] JMasar: A PDF? Don't be cheap, Jeff. Send me a hardcopy, okay?

[00:34:29] JMeyerson: I don't have hard copies yet. I'm sorry. I'll send you a hard copy when I have it. Okay, so like at an executive level, what does moving fast mean to you?

[00:34:41] JMasar: Hiring amazing engineers, and then having a different way of managing essentially any direct report. Meaning – Okay, so first of all, let's start with the very beginning. You start with looking for the right people. Typically, if you are a small company – Okay, there's the scenario. You're a small company, you're a small startup, just a bunch of friends, you just started a new business. You need to find amazing people, the best people that you work with. Sell them on the vision. Get them into the company, and then give them all the resources to be successful. What does that mean? Well, really, just removing any friction. I hate when small companies have restrictions on who can start a new culture code instance. This still happens, and it's so stupid. Like before you get into compliance, and you're not doing compliance when you have no product, everybody should have access to everything. And at that point, it's about you as the manager have the responsibility to figure out how mature is everyone, how autonomous they can be, and essentially start as a micromanager with every direct report. And every time they do something right, and you have no feedback, you take a step back a little bit and give them more space to do their work.

And at the end of the day, after a few weeks, after a few days, depends on how fast you're moving, the delegation, so called delegation, that a team can be like, “Hey, Jeff, can you implement CDN caching for me? Make sure that reCAPTCHA works, and don't scrub our analytics.” And the guy is like, Jeff’s like, “Yeah, sure, I can do that.” And that's it, right?

Here's the thing, many of the software best practices out there are talked about because big companies are talking about them. And they're talking about them, because it's good for hiring. But something that works for Facebook is not necessarily great strategy for a small startup. Let me be completely clear. Just you and your buddy, you haven't launched a product yet. You're just – This is the first week of you starting, writing a new software. You don't know if this is going to work out. The number one risk for the company, the number one business risk is that you're – Well, first of all, you're going to argue with your cofounders and really break up. That’s number one thing, statistically speaking. The second one is that you're not going to make something that people want, right? So we are coming back to the fact that you need to learn fast and iterate fast.

Why are you writing tests on your first week of a new product? Why do you have a staging environment, right? Why are you sharpening your database before you have your first users? Like here's the thing, it's not about the code style. Nobody cares about code style. It isn't about the code quality. The code quality doesn't matter. My aim is not to write amazing software. My aim is to have an amazing business. And the right thing for the business is to ship great qualities software fast. But great quality means it's great quality for the user, right? Software depth, engineering depth is amazing thing when you have it at the right place at the right time. And the beauty is – Well, here's the thing, perfection kills startups. Perfection kills companies and ideas, right? So you need to identify the places where you can afford not to be perfect. And you need to decide on what are the aspects of their product, which need to be amazing. Does that make sense? Do you know what I'm trying to say?

[00:38:16] JMeyerson: It makes complete sense. I agree. It's how I build products. There's a lot of heresy in what you just said, but I like it.

[00:38:23] JMasar: Companies like 37signals or Basecamp are doing have a great narrative about this. And it's aligned with the way I think about this, microservices versus monolith, right? Hybrid mobile apps versus native ops. Super cool new database that nobody heard off. First is Postgres. Like, really? Are you not using Postgres for your next project? Like give me the reasons, right? And most of the time –

[00:38:50] JMeyerson: I'm using Firebase.

[00:38:51] JMasar: Right. Cool. Good for you. As long as you're moving fast, I don't care, right? Here's the thing. Here's the thing. Think about the single point of failure argument. Next time somebody shows you their uber cool infrastructure – For instance, I hate Kubernetes. I don't think small teams should use Kubernetes, right? Let that sink in first.

[00:39:11] JMeyerson: Depends what you're doing. But okay, 90% of time, probably. I mean, but come one. It's not for most people. Like it's mostly –

[00:39:20] JMasar: Right.If you open up Hacker News, what is the number one thing you read? Oh, don't use jQuery? Guess what? Like try implementing the is visible method from jQuery without jQuery? Is it worth including a few kilobytes of JavaScript intervention if you're going to ship the product faster? So I want to think about all the best practices, so called best practices that you read on Hacker News, and revisit if it makes sense for –

[00:39:50] JMeyerson: That's a chapter from my book, Rethinking Best Practices. That is a chapter.

[00:39:55] JMasar: Yeah, lovely. I love that book. By the way, I was going to write that book. Fuck you, Jeff.

[00:40:02] JMeyerson: Were you really going to write a book how Facebook builds software?

[00:40:04] JMasar: Not necessarily Facebook writing software, but the way I like to look at software, yeah. Let me give you one more example. Something that I'm fascinated about. Why do people use cloud like EC2 or Digitalocean? Yeah, let's talk about EC2 for a second. Why do people use like scalable containers? Well, they use it because it's cool, right? But what is the underlying motivation? The underlying motivation is that you can scale your infrastructure dynamically whenever you need it, right? This is a super specific requirement that only some companies use.

First of all, when you're starting a company, you have no usage. So like why are you building a scalable backend? Like you don't know if the app is going to work out in the first place. But even if you are building a scalable backend, does it really need to be scalable? Scalable, meaning, that in 10 minutes time you add whatever. You double the capacity. Is it possible for your business that it can wait for two days when you add another physical server? Because if it does, then maybe it makes sense for you to either buy or rent out entire physical servers for super cheap and get better economics. And it's not about economics. It's about – You probably know, Heroku, right? Have you deployed anything to Heroku?

[00:41:17] JMeyerson: Oh, yeah. Entire companies.

[00:41:19] JMasar: Do you know how much memory you get on one Heroku? Do you know?

[00:41:23] JMeyerson: I don't remember. I'm sure it's minimal.

[00:41:24] JMasar: It's like 512 megabytes, right? So you're like, "Okay, I want to pay up for more, because we use Ruby. And Ruby is not great with garbage collection." So okay, I buy the largest, which gets generous, and this is me being ironic, 16 gigabytes of memory. And it costs like 500 bucks a month. Going back to Hetzner, or DigitalOcean and renting out an entire server. For that 500 bucks, I can get one server, which is going to have 500 gigabytes of memory. Not

500 megabytes, but a thousand times more memory. And now, Jeff, the ultimate question. Is it easier for you to develop brand new software at a computer which is scalable and distributed and you need to solve? You have many single points of failure. And need to solve the microservice communication? Or is it easier to host it on a single physical server with 500 gigabytes of memory? So even when you have a memory bloat, the server can tolerate it for a number of days, and you have a single point of failure. Your database, your Redis, whatever, whatnot, is hosted on that single machine. Like Heroku grew up, Heroku became a thing when it wasn't common for people to have a terabyte of memory in a server for super cheap, right? And why are small companies paying like 10 grand as a Heroku bill, and getting a worse experience?

So long story short, the moral of the story here is that startups are busy building their product. And they don't get enough time to talk about the way they develop software, right? And when companies become bigger and want to hire, attract talent, then they start blogging about how the cool ways or the engineering software. But at the same time, they're big companies. And their best practices changed in the meantime. So I would really question all the best practices. And if you're doing something just because that's the right way of doing them, doing things, or because that's really the best way for you to be productive.

[00:43:35] JMeyerson: Okay. Well, there're a lot more we could talk about. Elearly, we should wrap up. Anything you want to add? Or I'm sure we'll do that we'll do this again sometime. What else do you want to close with?

[00:43:43] JMasar: Yeah. So it's wwwbetteruptime.com. You will not regret it. No. On the serious now. There is an amazing book called *Refactoring UI* by the authors of all of Tailwind CSS. I encourage people to check it out. The smaller the company, the smaller the team, the more overlap in roles there is. And your full stack engineers, your backend engineers will be writing parts of the UI the smaller the team is. And the UI is going to suck when it’s written by somebody who thinks about the way the schema structure should be, the schema should be structured and not think about the UI. And I think you can write better UI. I think you can actually write better database schemas if you think about the end user, right?

And one book that I give everyone that starts here, it's called *Refactoring UI*. It's super visual. It's an easy read. You can literally go through the book because it's just full of illustrations. It gives you examples about what not to do and then examples about what to do visually. And you can become a better software writer by understanding UI better, because even though if you are not interested in UI, you will find yourself writing UI in many cases. And the ironical thing is that you will get better at designing backend if you understand the end user better and become better at designing interfaces.

[00:45:12] JMeyerson: All right. Well, yeah, it's been a pleasure talking.

[00:45:15] JMasar: Yeah, it has been fun. Thanks for having me over, Jeff.

[00:45:18] JMeyerson: Yeah. Jay, thanks for coming on the show. And best of luck with the company.

[00:45:20] JMasar: Thank you so much.

[END]