EPISODE 13

[INTRODUCTION]

[0:00:00.4] JM: A software consultancy solves problems involving management and software engineering. A large company might hire a software consulting company to give an external opinion on software architecture or on organizational structure. Sometimes, a consultancy is brought in to help integrate with a new technology or to do a major refactoring.

Scaling a software consultancy to meet the varying demands of clients presents a unique challenge. Software companies that make money from media or software as a service or advertising technology are primarily focused on scaling the technology. For a software consulting business, scaling and updating the team is arguably more important than any particular piece of software.

Rachel Laycock is the head of technology for ThoughtWorks North America and she joins the show to discuss how to manage and grow a large software consulting organization. It's a great discussion of culture and technology and how the nature of work is changing.

If you're interested in hosting a show for Software Engineering Daily, we're actually looking for engineers and journalists and hackers who want to work with us on content. This is a paid opportunity. You can make $300 if you get your show posted and you can go to softwareengineeringdaily.com/host to find out more. We want to make it easier for people to podcast about software. We want to be your backend for software engineering podcasting.

Also, the Software Engineering Daily Store is now open. If you want to buy a Software Engineering Daily branded t-shirt, a hoodie, or a mug, you could support the show and brand yourself. You can go to softwareengineeringdaily.com/store to find out more.

Now, let's get on with the show.

[SPONSOR MESSAGE]

[0:01:54.4] JM: Software engineers know that saving time means saving money. Save time on your accounting solution. Use FreshBooks Cloud Accounting Software. FreshBooks makes easy accounting software with a friendly UI that transforms how entrepreneurs and small business owners deal with a day-to-day paperwork. Get ready for the simplest way to be more productive and organized. Most importantly, get paid quickly.

FreshBooks is not only easy to use, it's also packed full of powerful features. From the visually appealing dashboard, you can see outstanding revenue, spending, reports, a notification center, and action items at a glance. Create and send invoices in less than 30 seconds. Set up online payments with just a couple of clicks. Your clients can pay by credit card straight from their invoice. If you send an invoice, you can see when the client has received and opened that invoice.

FreshBooks is also known for award-winning customer service and a real live person usually answers the phone in three rings or less. FreshBooks is offering a 30-day unrestricted free trial to Software Engineering Daily listeners. To claim it, just go to freshbooks.com/sed and enter Software Engineering Daily in the How Did You Hear About Us section. Again, that's freshbooks.com/sed.

Thanks to FreshBooks for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:32.2] JM: Rachel Laycock is the head of technology for ThoughtWorks North America. Rachel, welcome to Software Engineering Daily.

[0:03:37.6] RL: Thanks for having me.

[0:03:38.8] JM: I want to better understand the business of software consulting. There are prestigious management consulting groups, like McKinzey that are well-known for coming into a company for solving issues around management or workflows or accounting, and software consulting approaches those same issues or similar issues, like management issues, while also

looking into the software architecture of a company. When does a company typically hire a software consulting team?

[0:04:15.3] RL: I think, historically, they would bring in at a later point. After the McKinsey's and whatnot had helped them with their strategy and defining what the business problem is that they might want to solve, traditionally, a software consultant would come in when they wanted to execute.

As of the last few years, more and more organizations, traditional organizations have started to become what they're calling themselves technology organization. They're starting to bring in technology consultancies earlier which means organizations like ThoughtWorks and other technology consultancies perhaps put on many more hats including things around design and product innovation and also some strategy and organizational change as well. I think both consultancies like McKinsey have moved into the software execution world and we've moved into more of the design and strategy world.

[0:05:06.9] JM: What are the kinds of companies that bring in consultants? What's a typical set of problems they'll have that they want some external viewpoints on?

[0:05:16.0] RL: I would say up until very recently, it was generally large enterprise, organizations that would bring in consultancies. They already have some development and ITT or infrastructure teams of their own, but they need either extra speed or extra support or skills that they don't currently have.

This could be any industry. We've worked in finance, and media, just to name a couple. We've worked in many different industries, and there are obviously technology consultancies that work specifically in certain types of industries, but we generally will touch many.

Recently, what we found is is that some of that startups that might be in Silicon Valley, might not, as they've matured over current years, much like the enterprises they've created these monoliths that needs somebody to come in and review the architecture and support them and give them extra ability to execute. I think the market in terms of who's calling consultancies is starting to change as these organizations themselves become bigger enterprises.

[0:06:26.9] JM: I did a show a while ago about SoundCloud, and SoundCloud had brought in some people from ThoughtWorks to reframe their architecture and their architectural plan for going forward and they were going to a back-ends for front-ends model. I found that interesting because SoundCloud is — So many people think of as a startup today, but it had reached a point where they wanted external viewpoints, and they got them, and they found a way to move forward and some help with the early implementation side of things.

You've been at ThoughtWorks for 10 years, I think, or a little more maybe. How have the client preferences changed over that period of time?

[0:07:15.5] RL: Let me just think about — What do you mean by the preferences? Can you just —

[0:07:20.5] JM: The technologies have obviously changed. Maybe they need fewer people on a particular job or maybe they need more people. Maybe they're looking for different things, like today, they're looking to move to cloud or they want microservices. Whereas in the past, they wanted — I don't know.

[0:07:45.3] RL: That's just it. In the past, they wanted us to help them with our agile software delivery, not so much of what we did. We helped them either deliver — We would bring teams in the ground; analysts, QAs, developers, project managers and run entire projects for them. They had a very project mindset and we would come in and run a project and then we would be done.

Overtime, they want us to look at overall architecture or think about the problem that they're trying to solve and what would be the best approach. We call it a little bit like moving up the clock phase in terms of the decision point as to when they need to bring a consultancy in, but there are also other changes in terms of — Like you mentioned, there's the technology changes and there's things like continuous delivery which we were kind of early adapters and early writers about that topic, and it being heavily involved in that community for a long time, is that once you start to move outside of a pure execution on a particular, even if it's not greenfield, a particular project. It can be sometimes quite easily [inaudible 0:08:56.2] off and be a small executable piece of work.

Once you move into the continuous delivery space, you start looking at types of technologies like; do we want to move to the cloud? Do we want internal cloud? Are we going to use external cloud? Start looking at how you manage your data so that you can deploy continuously. You start having to make big organizational changes because you're not just touching one particular team, you're starting to touch, what's usually in an enterprise, a bunch of siloed teams; an info team over here, and a QA team over here, and a release management team over here.

What they require from us has changed in scope. What we've realized is that someone might call and say, "I want microservices, or we need with continuous delivery." We always try and take it back, "What are you trying to achieve with this?" because these are generally usually quite big changes, and microservices is just one particular architectural style that could suit you but it might not be the architectural style, and I think people often jump on, "Oh! That's going to solve our problem. That's going to solve our problem."

We try and at least identify where the clear challenges are. Then, the other — I guess the last thing is the scope of the things that work with our clients on has also changed much like the scope of technology that everyone works on has changed. We now get engaged in things like data engineering and streaming, because once you start to think about microservices and you might take an adventure with architecture approach, and then you start thinking about data streaming technologies, and you might need a data lake, and suddenly the scope of what you're actually building, and therefore the scope of the skills that a consultancy needs to have increases.

Things have definitely dramatically changed overtime from executing on a single project to twitching many, many parts of an organization even outside of just pure technology, working with the business side to understand the problem space, working with design, and just working with all the areas of our client's businesses as well.

[0:10:58.2] JM: As the technology trends change and you have newer things, like streaming, demands for streaming, and adventure in architecture, how do you scale the educational aspects of ThoughtWorks, because as you have more client demands for that stuff, you want to reeducate and retrain people working at ThoughtWorks? Then, of course becomes a draw for

people to come to ThoughtWorks where it's also this — There's this educational side of the company. How do you — Since you're in a managerial perspective near the top, how do you implement that; getting the right skills propagated throughout the company?

[0:11:49.0] RL: I won't lie. It is very challenging and more challenging all the time, because we're not a product organization where it's clear the skills that we need and the skills that we need to build, and we have to work with our clients to understand their need and then we're looking for patterns where many clients are asking for similar things and start saying, "Okay. This is an area that we need to invest in."

People have often come to ThoughtWorks, come to work at ThoughtWorks because of the talent that we've continued to hire, and people like, "I want to go work with these people, and I want to learn from these people." One of the biggest things you'll ever hear at ThoughtWorks say, their favorite thing is the people that they get to work with and things that they get to learn. That was always — Almost by osmosis, I guess, by who they work with.

Recently, we've had to start implementing a lot more training and get — I guess, grow up in terms of having a training organization and understanding how do we identify the capabilities that our clients need and which ones do we invest in. We're looking at those patterns and we've got now big chunks of educational pieces around things like data streaming and things like native mobile. We also try and create bootstraps to these things. Rather than reinventing the wheel, there's actually a ThoughtWork who's currently working on an open-source project called Wheel to try and bootstrap some of these challenges which actually allows some of our thought workers to get when they're kind of learning these kinds of things to get past the, "Okay. First, I have to setup a build pipeline. Now, I'll get to actually work on some code," is to try and bootstrap some of those things as well.

It's absolutely more challenging, and I think from a leadership perspective in one of the roles that I play is identifying and prioritizing where do we invest and trying to understand, "Okay. This is a clear area in the industry that's growing. We need talent in."

Like everybody else, we're also trying to hire. If we're looking, we have people within the organization that might already have these skill sets and then we were always been a big

proponent for pairing, so we can pair them with somebody who's learning after they've got through some basic training. Then, a bit more of that osmosis can happen quickly.

We're also heavily trying to hire much like everybody else, but the tele-market is also just so much more difficult and challenging than it used to be, where I think ThoughtWorks used to be an implorer of choice, because the agile, the continuous delivery, the types of thinking that we publish, the open-source work that we did. If you were really passionate about your craft, people would come to ThoughtWorks. Now, there are many organizations having that passion for craft. That's not our only selling point anymore, so it is very challenging.

[SPONSOR MESSAGE]

[0:14:53.4] JM: Spring is a season of growth and change. Have you been thinking you'd be happier at a new job? If you're dreaming about a new job and have been waiting for the right time to make a move, go to hire.com/sedaily today. Hired makes finding work enjoyable. Hired uses an algorithmic job-matching tool in combination with a talent advocate who will walk you through the process of finding a better job.

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

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

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

[INTERVIEW CONTINUED]

[0:16:19.7] JM: Getting back to the reeducation question. When a new technology comes out, like TensorFlow, for example, or Kubernetes, and you look at these and you're like, "Okay. Well, this is quickly becoming the de facto thing for container management or the de facto thing for building machine learning, deep learning technologies. How much time do you give it to — Do you sort of wait to hear from the clients that, "Hey, we really want somebody on TensorFlow," or do you just jump on it immediately?

I think part of this is probably you've also got this media, kind of this media side to ThoughtWorks, I think the ThoughtWorks radar probably helps with this, because then you both give an opportunity for some of the internal consultants to do journalism on their own, but you also have an outlet for other people in the company to come and read and maybe they can read about something like TensorFlow and they can say, "Oh, whether I'm going to be working on this at ThoughtWorks, or working on this on my own in the future, it will be useful for me to know TensorFlow." Then, once you get a critical mass, then you can teach the masses in a better fashion. I don't know.

Help me understand how you look at that, and so you can avoid being — because you obviously don't want to jump on a technology too early and then get burnt when people just drop off and stop using it.

[0:17:48.4] RL: Yeah, you've hit the nail on the head. It comes from many different places in terms of how do we prioritize and when do we kind of pull the trigger on mass investment on training. Absolutely, we're looking at what our clients were asking for. We're looking at what's happening in the industry, but we also rely on our people for a big chunk of this.

We have devops specialists inside our organization. We have frontend specialists. We have native mobile specialists, and the really give us a good read on what's happening, what's important, because usually you can tell by what they're passionate about. What's interesting about the radar — Radar was originally published as an internal tool, not an external tool. The reason it was published internally was for us to get a grip. As we became bigger and more distributed, what do we think is good and how do we kind of share our knowledge with each other? Maybe ThoughtWork is working in Brazil and some working in the U.S. and maybe they have similar challenges, but they're picking different technologies, and it's kind of became a

forum for them to submit their opinion and why one is preferred over the other and in what circumstance.

What we realized is that we started kind of publishing that externally and realized that people were really interested in that opinion because of the breadth of work that we get to do. It's good to have from a large consultancy that's very focused on software engineering to hear their perspective on technologies, and often people completely disagree with us. We're always very clear that that's our opinion based on our use and what our engineers think is interesting or useful and what we've actually used. We take that into account as well.

In terms of when do we pull the trigger on the mass aspect, that will generally come from our clients. In general, especially in the enterprises, they're not necessarily the first early adapters of a technology, and many of our people — You mentioned SoundCloud earlier. There's a lot of export workers in many different startups all over the place and we're still friends with them, so we can say, "Hey, you guys are using that. What do you think about it?"

We're taking all these different inputs in terms of what's happening in the industry, what are other people using that are more in that kind of very early adapter space, maybe in the startup world. Then, by the time the enterprises start kind of asking questions around that, we've generally usually got an opinion somewhere in our organization that can help us make a decision for them. Then, in terms of like when do we pull the trigger on the scaling is really when we can see that there are many people that are asking for this and needing this.

It's a fine balance, because there are certain things that we will just whole-heartedly invest in in terms of capability. Security is kind of just a low bar. You just want to make sure that all of your technologists understand some of the very basic security concerns, because that kind of filters through everything, but to something more specialized, like a particular tool, like Kubernetes — Sorry. A particular platform, like Kubernetes. That's going to take some time from us actually using it and implementing it on a client and having the ability for them to kind of go with adapting it in that instance. Also, leveraging our network in the early adapter space to understand the value of these tools.

One thing that we've always kind of stated at ThoughtWorks is we might not — We do innovate on something where we might not be the very, very first to come up with an idea or adapt a technology, but one thing we do help large enterprises with is adapting some of these things. We're like, "Oh, here's what we've learned from our friends over here. Here's how you can leverage that."

What's interesting now as I was saying before is that we're starting to now work with the Silicon Valley organizations who have kind of grownup who have some of the enterprise problems. We can now kind of pass the knowledge back in terms of what we've learned from helping entries clients tackle their monoliths and move to microservices and do things like continuous delivery.

Even though they've might have had amazing engineers right from the start, adapting new technologies — You know what it's like. You've got deadlines, there's things that need to happen. In a startup world, things move very quickly, so they don't always make the best architectural choices right from the very start.

[0:22:19.6] JM: You mentioned this increasing competitive employment market. It is retention becoming harder, hiring is becoming harder. Part of that is driven by the fact that if you're an engineer these days, you have so many options, and there are a number of online market places for work. There's sort of like Uber for work type of places. It seems like this is where a lot of the different labor markets in our world are going. It turns out that Uber was kind of a sign of things to come where people actually really like the on demand work style. They like to just pick up jobs and then do the work and then go do a different job or go sign off and go home. Whether or not people like Uber specifically, that modality of work seems attractive.

I think a place like ThoughtWorks probably had a lot of people where that appealed to before this became a platform that was enabled by technology. I remember I interviewed Jeff Norris from ThoughtWorks and he was talking about just — If I remember correctly, he was talking about the appeal of the novelty effect that you jump from project to project to project and you never just get on the same stale technology stack for three or four years, which people could take for granted today, but 5 to 10 years ago, there are a lot of people who worked for 4, or 5, or 10 years at a job. Now, there's a lot more switching around, a lot more 18 months, or 9-month gigs.

Has the culture of ThoughtWorks become more important around that? Because the way I think about it, when you have all these other platforms that people can use to get the sort of itinerant work, the sort of constant novelty that they might have gotten as a consultant 5 or 10 years ago, when they can get that from a platform, from an online platform, that kind of raises the stakes for ThoughtWorks. Maybe it makes you think like, "Okay. Now we need to really build a strong culture that people can't get anywhere else."

[0:24:48.8] RL: Yeah, you're absolutely right. The culture is really paramount to ThoughtWorks. It always was, but I think the selling point, like you say, was often your ability to continue change and keep learning new stuff. Also, so we attract to those kinds of people. I think the culture, the people get to work with an the relationships they build tends to be — It's still a very strong thing.

It's interesting, because I was just chatting to some engineers the other day, because I always kind of ask them, because sometimes over in leadership, we have big ambitious missions about how we could change the organization and the types of work that we could go after, and that's one part of our responsibility. The other part of that is protecting the culture, because it's so important to retention and hiring. Also just making it the great place to work that's kept people like me around for so long.

The culture is so paramount, and I ask one of the engineers, "Describe your perfect project to me." Sometimes you expect them to say, "It was this tech stack," or "We're building this really cool thing for a client." He said, really honestly, that stuff is great, but the key for me is the people that I get to work with and the team that we create and the clarity of what we're building. If the tech stack isn't the most exciting thing in the world, then I'm okay with it. You want to get challenged every so often, and that's one of the advantages that things will change.

I think when you're kind of a gun for hire, basically, in the Uber kind of world, you're not really — Unless it's a very long contract, you're not really going to get the opportunity to kind of really create a good team and a good culture that you could from a slightly longer running team. The fact that even though you do move around from project to project, you're still working with thought workers and it's always very clear, the attitude is very clear, our values are very clear. It

still feels like you work at the same company even when you can work in very different client environments.

You're absolutely right. It's a very challenging market and I think it's kind of a key to how we keep people, but it doesn't keep everybody.

[0:27:14.3] JM: Right. To what you said about you kind of build your group of friends or your social group through work often times. The further I get away from college, the more I realized, "Oh, there's really not this social anchor built in to every phase of life."

After college, I worked jobs for a while and then you have this interesting social group at every company you work at. Since then, going off and starting this media company and just kind of working remote, I do really miss that social cluster of — From different job to job, you have these different clusters of people and it's like, "Oh, there's Rachel. She's the character on the team. There's Joseph, he's like the silent one." You get this little dynamic with every little team that you have. Since going off and working on my own, I have kind of a remote team, but it's simply not the same as the office.

I have been thinking a lot about the value of remote versus office, because even though ThoughtWorks is something of a decentralized team, I know there are offices — I've been to the San Francisco to hangout. What do you think are the pros and cons of the remote versus in-person — Kind of this debate is really starting to heat up where you've got companies that are completely decentralized and are having a lot of success. There must be some cost to losing the centralization of offices.

[0:29:03.7] RL: Absolutely. You're asking the right person, because in my role, which is across all of North America, I really don't need to go and sit into any of our offices. I do travel. I do go and visit our clients, and I do check in with our people and all of those aspects of my job. The rest of the time, I could — I'm on calls, conference calls a lot of the time. I could just sit at home in my home office.

What I find is that I want to do that when I have work that I need to concentrate on. Even as introverted as I am, because I was a software engineer myself for a long time. I come in to the

office. I came into the office today and I really didn't need to. In fact, the office is incredibly quiet today, which is a shame. I like to have a little bit of a social aspect and see what's going on and feel like I'm part of something, because you can feel, at least for me, quite isolating. I know people that work in organizations like DigitalOcean where they're completely remote. I actually think you can definitely work effectively in that way. There's fewer distractions, and if you have the discipline to figure out how do you pass information along and make sure everybody is on the same page. There's techniques and processes that can allow you to do that, but you definitely miss on that camaraderie, that friendship building, and just feel like that culture aspect that's very human that even if you're an extremely introvert person, you might just want — Every so often. Maybe you only want that 20% of the time. People do want that some of the time.

I think some of these organizations, the key will be — And I think it's really big key to our culture, is we still have officer, and we do things like what we call a home office, say, every Friday, and so I can come into New York once a month on a Friday and meet thought workers who are working all over our east market accounts, some in New York and some not, and get to know more and more people.

Yeah, it's challenging, and I think having some way for people to still be able to come in. I know that that's another thing, but sometimes a little bit solved by people that go and work in these kind of shared office spaces that don't really belong to anyone, or you can hire a desk, so you can still have that. Again, you're not part of a team. You're not working with people who are on a team. I think my perspective is, is that with anything, it's not like one is the greatest way to do thing and one isn't. I think five years ago I have open client offices everywhere and realized that two other people that's incredibly distracting and hard to work in.

I think for a company to be successful and hire the best talent, they kind of need to have all opt-ins available. Yes, you can work remote. Yes, we have a small office, if you want to come in. At the end of the day — We have an office in New York. If every single person that worked for the New York office was in the office, there wouldn't be anywhere to sit. We don't have an office that houses everybody.

We have a place that most of the time doesn't get completely full, occasionally it does. In general, a place that if you are currently assigned to a client or maybe on that particular client,

you're not working on a Friday. If you want to come in and still feel connected to ThoughtWorks, you can do. I think the success for organizations will be in being to create the option for both.

[0:32:37.0] JM: Isn't that stuff around the open space — It's insane. It's insane how the entire software culture — Everybody moved towards open offices, and now we're starting to see all these pushback where it's like, "Did we just pick a huge mistake? It seems like people just like to work in quiet." I know personally, I just got constantly distracted when I worked at an open office.

[0:33:00.4] RL: It's absolutely true. I remember facing it myself, because when I first joined ThoughtWorks, which was a while ago, as a developer, this was the first time I worked at a company that was really big on pairing. At first, I said, "This is awesome! I'm getting to learn all the time." Then, I realized that I really sometimes miss with certain activities just to be able to focus.

The other thing was is even when I was pairing, if you're an bit open planned office, there could be just very random meetings going on and everyone stood around the story wall and talking about a story and it can get really loud. It can either get loud, because it's a heated debate, or it can get loud because people were telling jokes and you're like — It's get really annoying really quickly when you're trying to concentrate, or you have a deadline.

Yeah, I think we always, in our industry — I think, again, we're still quite a young industry. We just love to kind of swing from one extreme to the other and it's like one minute we're all in cubicles and it's dilbit and it's dreary and nobody talks to anybody and you're creating these little tiny worlds of your own. Then, you have all the issues of people not communicating. Then, you go to the other extreme which is like extrovert paradise, but a nightmare if you're the kind of person that just wants to concentrate on what you need to get done.

I think what we're realizing is that each of those decisions were made for good reason. It's like, let's get — As always, just let's get back to basics. Yes, we need a way to have a discipline around communication to make sure that a developer doesn't go off doing something for two weeks and then comes back and it's like, "Okay. That was totally the wrong thing."

We need something that solves that problem, but at the same time, you need to create an environment that everyone can succeed in. It is challenging, but I can see — This is even something in ThoughtWorks, we have little phone booths in our big open planned office. If you need to work very quietly or you need to take a call, you can do that. You get both.

Also, any consultant who's not currently assigned to a client or needs to be on the client at the time, can also work remotely. You can work from home if you want. That is absolutely a non-issue. I think that's really the only way to solve is to — As I said, is to create both, but we love to swing from one extreme to the other. I very quickly found that the open planned office was — If it was noisy and busy, just an absolutely nightmare place to work.

[0:35:32.9] JM: We've had a lot of listeners write into ask for advice on transitioning from engineering to management and you have made this transitions successfully. What are the most useful lessons that you've learned from this transition?

[0:35:50.4] RL: First of all, it's a very challenging transition to move from engineer to manager because it requires a very different skill set. One of the things I've realized overtime is it's not necessarily the best engineer in the team that's going to make the best manager, because sometimes the best engineer in the team is somebody who is just potentially a very — The huge introvert wants to very much focus on the tech. Maybe they're not super passionate about learning all of the people's skills.

I think the biggest kind of aha moment for me with moving into that role is you no longer — You're no longer solely responsible for how things are executed, and you have to learn how to get things done by getting other people to do them. It's a really difficult thing to do, especially if you're working with somebody who's potentially more junior or doesn't have experience in an area that you do and you just, "Hey, just give me the keyboard. I could do this in 20 minutes, or whatever," and you know that you have to wait for that person to learn and take their time, and that's just one example.

Also, in order to get people to execute on the things that need to happen that you're responsible for because you're the manager, but they are actually going to do a big chunk of the work, maybe not all of it. Maybe you will still do some coding depending on your role on where you're

at. There's an incredible amount of people, encouragement, and influencing. Thankfully, we've moved away from their command and control world of, "I'm going to tell you the 10 things to do and you can go away and do it and come back when you're done."

I think most engineering cultures, hopefully, aren't too much like that anymore. What you realize is that you end up being this kind of coach to all the people, because you're accountable for certain results, or certain product line, or whatever it may be. There are things that you need to get done and you might have a team of 5, 10, or 20, or 100 people that need to do these things and you need to figure out how are you going to create the right structures and how do you delegate effectively and how do you understand where people strengths and weaknesses are. Which stuff can kind of — Maybe they can fail a little bit on and which stuff they can't. It's completely different skill set to get you ahead around all of that.

I've had interestingly an expert worker engineer who's now, I think, a director. I forget which organization, but one of the startups in New York. He said, "Wow! I just had no idea how much people stuff was involved in your job." I said, "That is my job." I talk to people all day long."

What I'm trying to do is move along all the things that I need to move along, because when you're a manager in any respect, your job is not to be the best contributor. Your job is to create many great contributors and then also create future leaders who you see the potential to lead, not necessarily being the best engineer in the team.

[0:38:59.8] JM: I think you have to go from your IDE, being your portal, into productivity to email, or Slack being where you get your creative work done essentially.

[0:39:12.4] RL: Yeah. The real challenge is getting the feedback and measuring it and feeling like, "Do I own any of that success?" I find it weird when I go through my appraisals with my boss, and it's like you reflect, "What did I do last year?" I'm like, "Well, I can't really claim ownership for that, and I can't really claim ownership for that." Then, listening 10, 20 things that I moved along or helped people hit goals on that all kind of fed up to overall goals that I might have been accountable for. It's really — It feels so weird to be like, "Oh, I achieved these things. I helped other people achieve these things."

[0:39:52.2] JM: One thing I think about management is it's almost like — Sometimes it could be this problem of scheduling. We've had scheduling in computer science for so long. Nobody solved it, and it's like you're constantly trying to — Not only are you — You want to figure out the right tasks to move the ball forward across an engineering team, but you also have to figure out something that they don't really teach you anywhere. You have to read books to figure this out or talk to people unless you just have insane intuition, is you have to figure out how to ask people to do things and the right words and the right time.

If you ask somebody 4 p.m. on a Friday, "Hey, you're going to have to do this next week." Maybe that's the wrong time to ask them. Maybe you want to ask them on Tuesday morning when they're fully caffeinated and optimistic about stuff. It's like the timing is so important. There's all these little people issues that take adjustment.

[0:40:50.9] RL: It's incredibly nuanced. I had a new leader, manager, ask me, literally, this very question last week. They were like, "How do I get people to do stuff without telling them to do it?" I said, "You know, well, first of all, you're a consultancy. You've got a head start. What is your job as a consultant? It's more in a consultancy and it's just that word, you're consulting. You're not just doing. You're not just executing.

A part of any engineer, anyone that's hired in thought works is they have to learn some consulting skills. Consultant skills is very much understanding what is a client need or what is this person's need and what are we trying to achieve and how do we kind of map that up. It's exactly your point. When is the right time to have a meeting with that person? When is the right time to bring it up? Are they the kind of person that you need to spend a little time talking about what you're doing on the weekend with, or they're very kind of direct and they just want you to get to the point.

You kind of have to learn all of these things when you're a consultant. I think when you move into those roles at ThoughtWorks, I guess it's easy for us to identify people that could make good leaders or maybe the design group or technology leaders or leaders around capabilities, because it's like those people that can do it on the client, you're like, "Okay. You're kind of half way there. Now, you basically have to turn a lot of those consulting skills into what you need your team and the people on your team to achieve."

I end up telling people — I said to somebody yesterday, you need to read influence science and practice. I remember a few years ago, I was working on a client and it was when my leader, or manager, had told me to read this book. I had it on my desk and he looked at it and he was like, "Oh, dear." It's like, "That's what you guys are reading?"

He's a contractor, and he's like, "That's just like evil sales tricks." I'm like, "Oh — Uhm —" I was like, "Actually, it's really fascinating." He said, "It isn't." There's many books like that around how do you influence people, and consulting skills, and it really is — Like you say, you kind of have to go back and you need to learn those basics and then you have to start figuring out how to apply them. When people say that, they start to feel like dirty. They're like, "Oh, no. I don't want to learn all these evil sales and consultant skills." I'm like, "Well, you're using it for good, no for evil, if that makes you any better about it." You're dealing with people and you need to understand what motives them, or the time of day that you ask them, or the type of activity that you ask them to do will determine whether it's successful.

You need to make sure you ask the right people to do the right things, of things that — Finding out what they're passionate about. All of these different aspects to it. I think you get the instincts for it after a while when you kind of — You practice, you learn the theory, and then it seems to come naturally, but I think even after a while, you meet difficult people who have difficult challenges and you have to think about, "How do I motivate this person and how do I make them happy, but also ensure that they get the job done?" It's very, very different from being the best engineer on the team.

[SPONSOR MESSAGE]

[0:44:12.4] JM: For years, when I started building a new app, I would use MongoDB. Now, I use MongoDB Atlas. MongoDB Atlas is the easiest way to use MongoDB in the cloud. It's never been easier to hit the ground running. MongoDB Atlas is the only database as a service from the engineers who built MongoDB. The dashboard is simple and intuitive, but it provides all the functionality that you need.

The customer service is staffed by people who can respond to your technical questions about Mongo. With continuous back-up, VPC peering, monitoring, and security features, MongoDB Atlas gives you everything you need from MongoDB in an easy-to-use service. You could forget about needing to patch your Mongo instances and keep it up-to-date, because Atlas automatically updates its version.

Check you mongodb.com/sedaily to get started with MongoDB Atlas and get $10 credit for free. Even if you're already running MongoDB in the cloud, Atlas makes migrating your deployment from another cloud service provider trivial with its live import feature. Get started with a free three-node replica set, no credit card is required. As an inclusive offer for Software Engineering Daily listeners, use code "sedaily" for $10 credit when you're ready to scale up. Go to mongodb.com/sedaily to check it out.

Thanks to MongoDB for being a repeat sponsor of Software Engineering Daily. It means a whole lot to us.

[INTERVIEW CONTINUED]

[0:46:07.1] JM: In terms of being malicious or manipulative, people would never say that about the communication that you use to make an API call. If you made an API call or if you wrote some code in XML these days, people would probably like, "Why are you writing that in XML? You should write it in JSON It's just could be friendlier to work with JSON." The same is true for interpersonal communication. You don't want to ask people to do stuff in the wrong way. Just ask them to do it at the more timely, more efficient fashion.

Anyway, specifically from the point of view of managerial stuff around consulting, how do you think about getting consultants and the full-time employees at the companies where a consultancy is working with? How do you get those two different groups of employees working harmoniously together?

[0:47:11.3] RL: it can be very challenging, because sometimes the culture of where these employees work is very different to our culture. You might have — They've potentially worked in a very traditional, non-agile waterfall type of style for a very long time. They used to putting their

headphones on and working alone. They're potentially being measured by weird metrics that cause weird KPIs that might cause problematic behavior.

There's a lot to unpack, and so this is why we end up with kind of layers of support on even engineer teams on the ground who were delivering something that's fairly straightforward in terms of what the need is. If we're delivering alongside client developers and client analysts, or client product people, or even contractors, figuring out how to get the team to gel can be a very challenging aspect of that.

Sometimes it goes really well, and sometimes it doesn't, because people are people, and ThoughtWorks is known for our technology skills, and I think everybody is always learning the best way to work effectively with other people.

A role that I've often played when I've been a leaders on those accounts is helping our consultants see things from other perspectives. Maybe their excitable, young, very passionate, very pedantic about something in the codebase and they might be working with a contractor, or a client developer who's very experience, but not experienced in TDD, for example, or in event-driven architecture, or something that we're implementing. You have to be careful of the new instance there. I think the trick is you need to make sure you don't get people on the defensive and trying to assure people that you're not here to steal their job. You're here to help them. You want them to learn, but you don't want to patronize them. It's a very challenging aspect to it, and that's why we have specialists in our organization who look at things like organizational change management.

I always say, and he says as well. He says, "In ever engagement that we have with a client, there is always an aspect of that." Maybe at the very low level, just how do we work effectively with their development teams, but it can go all the way up to how do we restructure their organization around a way that it's going to be more effective for the business. It's almost like different levels of needing to do that kind of change management as supposed to you do it or you don't.

Very much just the teams on the ground, it can be around coaching our people and just basic consulting skills and keeping up-to-date and understanding the perspective of where the people

are coming from that they're to essentially change, because we've been brought in to do something that potentially very different to what they're done before, but also even the way that they're doing things.

[0:50:26.6] RL: I don't know to what degree you can talk about this, but I'm actually really curious about the economics of a consulting company. Could you talk at all about the sales process, or how the client engagement works, and how you come to a price of a specific engagement?

[0:50:45.5] RL: Slightly out of my field of expertise, but I can certainly try and answer it. We have some fairly default engagement models, especially if it's a fairly traditional; come in, help us with delivery, help us setup continuous delivery. You might need to do a little up scaling of our teams while we're at it. We have some fairly standard engagement models which have standard pricing associate with them and we have these bunch of spreadsheets that operational and finance and salespeople work through to identify what's the best leverage model. Leverage is a big word you hear a lot in consulting. How close are we to kind of the pyramid shape of having fewer senior people and more junior people. That's just one engagement model, and it totally depends on what the client is asking us to do.

We have some engagement models which require — Because it's an architecture assessment, for example, lots of senior people. The profitability of those engagement models has all kind of worked out beforehand and we obviously have a — We have what we call a deal review system where very senior leaders, either from a technology space, a product space, in all the different areas, mobile, whatever it may be, will review these things as they come in to make sure that they also make sense. That we can deliver on the things that we're going to deliver on with the people that they're suggesting that we deliver with.

It goes through legal and all of those logistical things need to occur. There's a big review process. There's also back and forth and negotiation with clients. I think in terms of how do we engage and what our engagement models, I expect they're very standard, very similar to all the consulting organizations. I think how do we make money is really on how utilized our people are. We don't want them, what we call sitting on the beach, because then they become an operational cost. How much we can keep costs low, and it's pretty standard for a consulting

organization. I don't know if that answers your question, but that's kind of my view of the world from where I sit.

[0:53:02.6] JM: Yeah. Fair enough. How do you see consulting changing in the next 10 years?

[0:53:08.1] RL: That's a really great question, because I recently talked to, I think it was a forester analyst. In the past, the ideal consulting model was always the pyramid. As I was saying before, you have some of the very senior, and then you have a few people in the middle. The more junior people you can get on the ground, the more kind of leveraged the team is, the better profitability.

Now, the problem is, is that what's changing is both the client need, the breadth of knowledge that's seated in the industry, the breadth of things that we're doing for our clients, everything from frontend development all the way to kind of data or engineering work, data science. We're growing our capabilities much like they need to grow theirs.

What's changing is is that it gets very complicated to have just simplistic leverage model when you need lots of different types of capabilities at different points in the engagement. Some of our engagement models have decided to change where we think a lot, and I know consultants is already through this, about how do we kind of build in time for specialists' skills if needed, or bring in our mobile lead to kind of define the architecture for the mobile so that the rest of the team can kind of move forward as we kind of build out those skills.

What's actually really changing is instead of it being a pyramid, they're predicting that it will turn into more of a diamond shape where you still have a small number of very senior architects and leaders kind of running the account, or the engagement, and you have a small number of kind of generalist consultants. Then, in the middle, you have this kind of specialist group who deal with the different specialist around the organization.

What's interesting is for ThoughtWorks is we've always kind of touted, "You need to be a generalist. You need to be able to do full-stack." The T-shaped people, as in people that can do full-stack with little specialists here and there, the easiest — Not just for us to help our clients, but also for clients for their developers, that's what they should be going after too.

What full-stack is 10 years ago is very different to full-stack now. Full-stack was, “Okay. You’re a Java developer. You can do Oracle and SQL and you can do JavaScript and jQuery.” That’s a very simplistic model, but it wasn’t that far off. Being a full stack developer wasn’t the hardest thing in the world.

These days, it’s like which do you mean, like a JavaScript kind of full-stack. When you say mobile, do you mean native, or not native? Do you mean android of Apple? These things are just very, very different. I have one of our mobile leads here, you would say she’s a generalist mobile developer, because she can do all of those different things. She, in her prior — Earlier in her career, she did Java and whatnot. She is a generalist with specialist skills, but could I throw her on a data engineering project? I don’t think so. I think the consulting game is having to adapt to that and change our engagement models and we all are in order to meet those needs that our clients have.

[0:56:24.2] JM: One exciting trend that I see is the rise of these high level APIs and just how good they’re getting. You can realistically imagine a future in 10 years where, basically, everybody is just like writing a little bit of JavaScript as glue code between really high level APIs, image recognition APIs. Yeah, Amazon has got some API that shuttles all your data into this thing that’s really easy to work with and you can visualize it all in one place and you just write a little blab of JavaScript to shuttle it to the next place. You’ve got very easy to work with drag and drop UI configuration.

It seems like things could get so much easier to work with that it would almost become like more of a design challenge in the future. I think that could be really interesting.

[0:57:17.3] RL: Yeah, I agree with that, and I think sometimes when you think about that, it’s like, “Isn’t that from 10, 20 years ago when it was like just plug and play enterprise service busses?” Immediately, you just kind of go, “Oh my God! No. That can’t be a good thing.” We’ve definitely matured in what these APIs and these plug play ability is very different to what it was.

I think in that kind of future, what the role of a developer or a software engineer will be could be very, very different, especially when you start thinking about the impact of machine learning

overtime. It's hard to predict and people that are starting out now, and I do ask this sometimes, "Should we be learning to be like a software engineer now, because will we even have a job in 10 years?" My perspective is, is like, "Oh, there's going to still be a lot of software hanging around in 10 years." It could be a while before that happen, but yeah it's interesting and the changes in technology just way surpass even what I thought when I got into the industry, which is why I thought it was so interesting. I said, "This is a really new and a very changeable industry, and there's a lot happening." It's just exploded. Yeah, it continuous to be very interesting to see what happens.

[0:58:34.8] JM: Rachel, I want to thank you for coming on the show. It's been a great conversation. Really interesting.

[0:58:38.0] RL: Thanks for having me. I've really enjoyed it.

[END OF INTERVIEW]

[0:58:42.3] JM: Thanks to Symphono for sponsoring Software Engineering Daily. Symphono is a custom engineering shop where senior engineers tackle big tech challenges while learning from each other. Check it out at symphono.com/sedaily.

Thanks again to Symphono for being a sponsor of Software Engineering Daily for almost a year now. Your continued support allows us to deliver this content to the listeners on a regular basis.

[END]