EPISODE 438

[INTRODUCTION]

[0:00:00.3] JM: You've heard the phrase “every company is becoming a software company”. An insurance company is now supposed to turn into a software company that sells insurance. A clothing retailer needs to reinvent itself to be able to build software to manage the production and distribution of its clothing. Software applications provide so much leverage to an organization, it seems smart to develop in-house software teams to build those applications, but does it really make sense? Is there a better alternative to all of these companies becoming software companies?

In the 90s, outsourcing was a common solution to this problem. If you didn't have software expertise within your company you would hire a large consulting firm. These firms would often hire inexperienced offshore developers and the resulting code quality was not so great. Because of the bad experiences of the first internet boom outsourcing, companies became more cautious about outsourcing their engineering work, which led to today, where the standard is to hide your own software team.

The world has changed in ways that have made outsourcing a more viable solution. Programming best practices are more widely understood. There’s the international community of software engineers who share information on places like Stack Overflow, Quora and Twitter. Off-the-shelf collaboration tools make it much easier to communicate the requirements of a project to a team of developers.

Gigster is a company that is working to optimize the engineering of software projects. Large enterprises come to Gigster to build new projects from scratch, whether that project is a marketplace, a mobile application or a machine learning model.

Roger Dickey is the CEO of Gigster and he joins the show to describe how Gigster works and why it often makes sense for companies to focus on their core competency and outsourced software engineering.

Like this show about Gigster, some of our most popular episodes of Software Engineering Daily describe how leading software companies are built and organized. We've covered companies like GIPHY and Netflix, DigitalOcean and Stripe and many others, and you can find these easily if you download the Software Engineering Daily app for iOS or android. You can hear all of our old episodes. They're easily organized by category, and as you listen, the Software Engineering Daily app gets smarter and recommends you content based on the episodes that you're hearing. If you don't like this episode you can easily find something more interesting any time by using the recommendation system.

The mobile apps are open sourced and github.com/softwareengineeringdaily. If you're looking for an open source project to hack on, we would love to get your help. We're building a new way to consume software engineering content and we've got a bunch of different projects that you can contribute to if you're looking for open source. We've got the android app, the iOS app, this recommendation system backend, this web frontend and more projects coming soon. People are talking about a chat application that is seems to be coming to fruition. If you join the Slack channel, you can join that conversation. Our Slack channel is findable by going to softwareengineeringdaily.com and you can also send me an email at any time if you're interested in getting involved. I will point you in the right direction. I'm jeff@softwareengineeringdaily.com.

With that, let's get on with this episode; Gigster.

[SPONSOR MESSAGE]

[0:03:35.9] JM: When your application is failing on a user's device, how do you find out about that failure? Raygun lets you see every problem in your software and how to fix it. Raygun brings together crash reporting, real user monitoring, user tracking and deployment tracking. See every error and crash affecting your users right now. Monitor your deployments to make sure that a release is not impacting users in new unexpected ways, and track your users through your application to identify the bad experiences that they are having. Go to softwareengineeringdaily.com/raygun and get a free 14-day trial to try out Raygun and find the errors that are occurring in your applications today. Raygun is used by Microsoft, Slack and Unity to monitor their customer-facing software.

Go to softwareengineeringdaily.com/raygun and try it out for yourself.

[INTERVIEW]

[0:04:41.7] JM: Roger Dickey is the CEO of Gigster. Roger, welcome to Software Engineering Daily.

[0:04:46.3] RD: Thank you.

[0:04:47.2] JM: Today we're talking about your company; Gigster, which does outsourcing, contracting for large enterprises and basically it's a way for enterprises to get applications built for them. I want to start by talking about outsourcing from a historical perspective. In the 1990s, outsourcing got a bad reputation. What has changed in the last 20 years that has made outsourcing a better value proposition?

[0:05:21.6] RD: Great question. I'm familiar with industry, but also new to it. So I of course wasn't in it, to caveat this by saying, I wasn't in the industry in the 90s, but I think what we saw in the 90s was this explosion of off-shoring talent. As with anything else that causes — That it crosses cultural boundaries, I think there were growing pains. There were natural growing pains. Talent overseas is pretty incredible. Some of the best engineers at work to have been Indian or Russian based engineers, but there are a lot of challenges on the interface between that talent and clients.

I think you saw companies that can afford to pay for the best have kind of less of an interest in consulting firms that compete on cost. Many companies that need something done right, they're not necessarily go somewhere that they can get it done a third or a fourth of the cost if that sacrifices the quality of the end product, the maintainability of it, the sort of total cost of ownership in the same sense that you have that for a lower quality versus a higher quality car and just the general interface.

I think what you see people often do is they'll hire agencies as sort of an indirection layer. So they'll hire a US-based creative agency that can conceptualize the product and then that agency

may outsource to someone else. I think now that overseas talent has kind of equalized in cost with even some US-based talent, we're starting to see a lot less of that happen especially as companies care more about quality.

[0:07:02.3] JM: Do you think it has anything to do with the fact that the tools we use these days are higher-level, maybe a bit friendlier. It's easier to piece together the building blocks of an application today from, say, AWS and Twilio than it was back in the 90s when — I don't know what you were using in the 90s, but I guess like Java J2EE or something.

[0:07:34.3] RD: Yeah. Certainly, higher levels of obstruction and all these infrastructure services have been hugely helpful. I think that kind of democratizes who can do the consulting, and it raises the average quality level for sure. I think there is always a frontier that's hard for companies to access that's new. These days it could be — It could be your augmented reality app, a machine learning app, some sort of data science project. I guess it's probably still hard to outsource that, but mature technologies are certainly getting a lot easier to handle.

[0:08:09.6] JM: You hinted at it there, but how do the contracting needs of a large company look like? If you're IBM or eBay, what kinds of — You're a software company. So what are the things that you want to outsource?

[0:08:28.9] RD: It's interesting that you mentioned both of those, because those are actually both our clients. We've got about 40 enterprises what we work with. Honestly, I had the same question myself when I was founded Gigster. We thought we'd be helping non-technical firms outsource. Maybe if you look at the Fortune 500, right? How many of these 500 companies are actually technology companies? Maybe less than 10%, less than 50 I would say. It could be 30.

You have a lot of firms that are fundamentally technical, which represents a large market if you're bringing technology it. What we've seen is that, like the examples you gave, even the IBMs and eBay's of the world have a need. I think it boils down to, let's say, three areas. One is quite simply the roadmap. The roadmap grows faster than their technology department can consume it. You have people who need more bandwidth, and we come in and provider extra bandwidth sometimes. We simply take projects off the roadmap that they don't have the resources to execute themselves.

There's always some new executive that has some initiative and it's the most important thing in the world and it's more important than this other executive's initiative. There could be a political battle internally for resources or access to IT. We come in and essentially obviate any political battles, because we'll give anyone technical resources. I think that's one.

Another example is speed. The project needed for IBM had to be completed in two and a half weeks and they weren't equipped to do that internally. We basically came in and delivered that for them and we spun up really fast. We spun up a team like an AWS instance just within days and got that done.

The third thing we see is talent that they just don't have internally. I was talking to a company today, a German insurance company called Allianz and they have a ton of engineers internally, but there are certain types of talent that they don't have, so they might not have machine learning engineers. We'll come in and give you a team full of engineers like that if you need them.

I would say it's speed bandwidth or access to resources they don't have, which would be the case with Allianz if we worked with them and it's the case with eBay and other one of our customers. They need machine learning engineers.

[0:10:43.0] JM: There is a narrative that every company is becoming a software company. If you're an insurance company, you're actually a software company that happens to sell insurance. I'm wondering to what degree you think that is the case, or should an insurance company that has increasing software needs, should they be scaling up their outsourcing, for example, and just focus entirely on their core competency of actuarial tables or probability of car accidents or something like that.

[0:11:22.9] RD: Yeah, this is something we talk about a lot and it's a great way to frame a common question that businesses have. I remember hearing from somebody that Starbucks's new CEO is telling people that Starbucks is a software company, which is interesting, because I imagine none of their customers would think of that way, but they do have software products

that power supply chain logistics, customer loyalty through mobile applications, etc. I think you could call a lot of companies software companies.

I do a fair amount of angel investing. One thing that I look at with companies is — Obviously, I look at whether or not technology is the core hard thing they have to do to win. If it's not, I have no business investing, because I'm not familiar with other types of businesses.

I think if you're going to call something an X-company, if you're going to call something a technology company, then it has to be the case that technology is the core thing they need to do to win in the market. I think some businesses that are calling themselves software companies, maybe that's not actually true. Maybe the executive team at that moment in time is excited about technology, so it's sort of a rallying cry to get the company more going down that path.

I do think many companies are software companies. Any company that acts on data, for example, an insurance company, is a technology company, because technology acts on data and if their competitors have more technology than they do in that realm then they won't succeed. You have other companies that are really more about the brand. Something like Nike. Granted there's a lot of supply chain, but also their brand is very important.

I think it really depends on the industry, but you do definitely see — I think you see a lot more companies these days that are technology companies and behooves really every business in the world to have their own custom technology.

[0:13:17.4] JM: I've outsourced a lot of different projects that I've built, because sometimes I like to just architect out an idea or do some wire frames, and I don't want to write the MVP. I just want to outsource it. I think the needs of an enterprise, it's often the same thing. They've got a new project they want to spin up and they outsource it to something like Gigster. Then they have an MVP that they can work with and they can hand it off to somebody at the team within their company that can manage it going forward. I do want to talk about that with you in a moment. I want to talk about the process that if a large enterprise is getting something built, how they work with you, with Gigster specifically. Talk a little bit more about outsourcing fundamentally and how it might look in the future.

Outsourcing is obviously getting easier. It's getting good. Something has changed relative to how outsourcing worked 20 years ago, where it has become quite an appealing value proposition. You know more about the fundamental economics than I do, so I'm really not sure. Could you imagine a world where the insurance company just says, "You know what? Our competitive advantage is we're really good at doing sales and marketing and getting customers to come in through the door and we just kind of get data and maybe we get most of the — Most of our applications are outsourced. Maybe we have a couple of engineers who work internally to sort of manage projects. They're more project managers, but they outsource all of their engineering work." Can you imagine a world where that might happen, where it's actually more economical? Rather than if you're All State trying to hire software engineers and figuring out an equity compensation package that's really appealing to them. You just do all of your work through outsourcing. Can you imagine that kind of world?

[0:15:22.5] RD: 100%, and I think you were touching on that with your previous question, but I'm glad you brought it back up, so I don't think I really address it. It's one of the reasons we started Gigster, honestly. The vision for us is to be a central engineering department for every business in the world. We call it the world's engineering department too, and that's exactly the idea that technology is essential for every business, yet most businesses don't really have easy access to it.

I think it's the case that great software developers are far less expensive to a business than not so great software developers. The same total cost of ownership sense that we talked about before. We've all talked about 10x developers and heard that idea. Well, a 10x developer doesn't cost a company 10 times more than a 1x developer so to speak. As a business, like an insurance company that doesn't have tech as their core competency, how do you attract a bunch of 10x people? They don't want to work for you.

I think what the future might look like is that there's one central or several central entities that do most of the engineering, and the great talent is collocated in those central entities. Companies have a thin layer of technologists, but they're mostly focused on their brand, their core value proposition. Maybe the insurance company really has one algorithm that differentiates them from every other insurance company. Sort of like the secret source of special ingredient that a

restaurant might have that its entire business is based on. After that, they don't really need to bother themselves with the details.

I think we're already seeing that. We're seeing that with a lot of our customers. Even before we got there, they had other companies that build most of their technology. I can't name specific names, but a lot of customers don't really have much technology. It's sort of a skeleton crew internally.

It makes sense due to market trends. The best engineers want to work with good people on interesting problems and there just aren't that many at some of these companies. I think you're seeing the best engineers go build frameworks. You're seeing them go join organizations like Google and Facebook. There's definitely a rich, get richer type of dynamic which really hurts if you don't have great engineers already. Building that culture, understanding how to work with engineers simply won't be the competency in a lot of these companies.

[0:17:40.2] JM: Yeah. We'll get a little more into that later on, but let's get people understanding for how Gigster works. Describe the process of an enterprise coming to work with Gigster.

[0:17:55.0] RD: We typically — Somebody will reach out to us. What we try to do very quickly is understand at a high level what the problem is they're trying to solve with technology rather than get down the rabbit hole of a conversation about a specific mobile app or website they want to build. Then we go from there.

Sometimes what we'll do is we'll do — If their requirements aren't clear-cut upfront, we'll do a scoping engagement where we'll come in, we'll help them articulate what they want to build. We'll create a spec document and we'll give that back to them. They can take that, implement that internally. They could give it to a different development solution or they can hand it right back to us, in which case we'll build what they need. If they come to us with a specification already made, then we can go build that.

The way our process for development works is we need to have a clear-cut spec upfront. We don't do agile work. Everything we do is closer to waterfall. It's fixed price, scope-defined upfront. We allow for some reasonable flexibility during the process, but we do require an

upfront well-defined scope. Then we manage the process through a series of milestones with the client.

Our freelance developers aren't paid by the hour. They're paid by the milestone that they complete, and then we're paid by the customer also by the milestone. None of it is hourly based. It's not a staff augmentation model. We won't give you individual developers. We only give you an entire team. The reason we do that is if we understand what you want upfront and we have full control over the team and how the freelancers operate, we can actually optimize what happens under the hood.

The same sense that, with Uber, you don't — It's not a marketplace where you meet specific drivers and hire them. You push a button and the app finds you the best person. Maybe someday the app will give you a self-driving car or it will just teleport you. It will do the most efficient thing to get you to your destination. We think of putting a software team together and delivering a consulting service the same way.

[SPONSOR MESSAGE]

[0:20:02.8] JM: DigitalOcean Spaces gives you simple object storage with a beautiful user interface. You need an easy way to host objects like images and videos. Your users need to upload objects like PDFs and music files. DigitalOcean built spaces, because every application uses objects storage. Spaces simplifies object storage with automatic scalability, reliability and low cost, but the user interface takes it over the top.

I've built a lot of web applications and I always use some kind of object storage. The other object storage dashboards that I've used are confusing. They're painful, and they feel like they were built 10 years ago. DigitalOcean Spaces is modern object storage with a modern UI that you will love to use. It's like the UI for Dropbox but with the pricing of a raw object storage. I almost want to use it like a consumer product.

To try DigitalOcean Spaces go to do.co/sedaily and get two months of spaces plus a $10 credit to use on any other DigitalOcean products. You get this credit even if you have been with DigitalOcean for a while. You can spend it on spaces or you can spend it on anything else in

DigitalOcean, and it's a nice added bonus just for trying out Spaces. The pricing is simple; $5 per month which includes 250 gigabytes of storage and 1 terabyte of outbound bandwidth. There are no costs per request, and additional storage is priced at the lowest rate available, just a cent per gigabyte transferred and two cents per gigabyte stored. There won't be any surprises on your bill.

DigitalOcean simplifies the cloud. They look for every opportunity to remove friction from a developer's experience. I'm already using DigitalOcean Spaces to host music and video files for a product that I'm building, and I love it. I think you will too. Check it out at do.co/sedaily and get that free $10 credit in addition to two months of spaces for free. That's do.co/sedaily.

[INTERVIEW CONTINUED]

[0:22:20.7] JM: I like the milestone model. I think it keeps incentives aligned so you don't have engineers purposely taking a long time to finish a milestone. They just finish the milestone as quickly as they can and then they get money based on that.

How do you figure out how to agree on a price upfront and how do you prevent scope creep? Because a lot of people that you're probably interacting with, they're used to Agile. I like that you own the term waterfall. I also like that you own the term outsourcing. I don't see either of these things as bad words. They've been maligned due to just bad experiences that people have had. Tell me more about how you figure out what the price is going to be. When a company comes to you and they're like, "Hey, we want to do project X," and you're like, "Okay. It's going to cost X, and here's the milestones that we're going to break it down into," and how do you prevent the scope creep?

[0:23:21.0] RD: This is one of the areas that our core technology comes in. In a sense, we've mapped the software genome, so we understand when a new project comes in. We tag it with different details and we understand what features it's made up of. Project has various meta variables, like what technologies that you use is — Whether it's a rushed job or not. What platforms does it have to integrate with existing cost for technology? Then it has a list of features; photo upload, a settings page, a landings page, a feed, comments, likes. Each of

these could be different features in an application, and we tag a new project that comes in with these features.

What that allows us to do is find similar past projects, and we've done thousands of projects for clients, so we can actually find very similar projects. We can look at data and how those went, and this data comes in from GitHub, from JIRA, from Trello, from Slack. Whether it's communication project management or source check-in, we aggregate all these data and we can get an understanding of how the project went, use that to predict the optimal time and price for future projects, and we've gotten to a point with that where we're actually very accurate. We obtain our margin, probably 98%, 99% of the time on projects, which is pretty stunning, but it's also something that's structurally possible given our model.

[0:24:39.5] JM: How much do you talk about that margin? How much the developers are getting versus how much Gigster captures. Is that confidential?

[0:24:48.1] RD: Yes, it is. It also really just depends on the project. The margin is very low on some projects, because we need to redo work or we have a lot of internal company resources, like customer support that have to heavily support our freelancer base. In other cases, it ends up being higher if we're able to reuse code assets across projects. It's really highly project dependent, but the reality is what we do for our freelancers is we take on all that risk so they can just essentially write code and get paid. We try to make it the easiest possible experience for them.

[0:25:20.4] JM: There's an interesting model for how the client is interacting with the developers that are writing the project. Let's say I am a company, a big enterprise and I come to Gigster because I need this machine learning application built. I say I've got all these customer data and I want a machine learning algorithm that's going to predict whether this customer is going to churn or not. Here's kind of the data. Here's what I'm looking for, and the enterprise that you're dealing with will actually interface with a project manager, and then the project manager will manage the engineers on the Gigster side. The engineers actually don't ever interact with the enterprise that's coming to Gigster as a customer. Is that accurate?

[0:26:10.8] RD: That's right in most cases. Our engineers, one of the reasons they work with us is they don't want to interface with clients. They could join another freelance network and they could work directly for clients in a staff augmentation type of arrangement, but our engineers like us because we have that indirection layer. We protect them. We make sure they're paid on time. The project manager is typically a freelancer as well, although we do also have internal resources that oversee the whole project.

Really, it's an entire fully functional freelance team where all of the development resources are kind of been directed from the customer with that project manager. That's always been our model, and it's a bit essential to have that kind of quality control that manage service layer to make sure that we're able to hit margin given that we're guaranteeing a fixed price.

[0:27:04.1] JM: Why is that so important to hitting the margin? I don't quite understand that.

[0:27:08.0] RD: Because if the client was working directly with the developer, what we might see is the client randomizing the developer, scope creeping the project. Can I get the developer to do extra work that wasn't part of the original project plan? Great developers don't want to also have to do project management and debate feature creep with clients as they're building a project.

Project managers are this line of defense in a way, where if the client comes in and tries to change their requirements or the client doesn't like a deliverable, the project manager can kind of insulate the development team from that.

Our project managers have to push back sometimes. Sometimes it's the opposite. Sometimes we try to over-deliver for the client, but it's nice to have that layer in to make sure that someone's watching budget.

[0:27:58.7] JM: It makes a lot of sense. The project managers or project managers at Gigster, are they managing multiple projects concurrently?

[0:28:08.1] RD: In some cases, they are. We have project managers that manage up to, I would say, 8 projects at once, something in that range. We have others that manage one or two

projects and others that are between projects. Maybe their full-time gig has been busy. They've got a startup or a small business or a family issue on the side. We don't mandate that they work any amount. As long as they're able to do a good job, they can be on as many projects as they like.

[0:28:34.1] JM: That's one thing I was interested in, is how Gigster itself scales as the number of project scales. If you had a linear increase in the number of projects that were on Gigster, I guess the number of project managers or product managers that you would add would scale sub-linearly, because every additional project manager can probably manage multiple projects. You probably get some economies of scale there. Can you talk more about the economies of scale that you get as you get more clients coming in the door?

[0:29:15.2] RD: Absolutely. I'd say some are on product management. Others are around development and quality assurance and other areas. The way we think of — I'll kind of zoom out a little bit and then I'll talk about it and I'll answer this directly. The way we think about kind of work automation, I'll say, or this feature of work, concept of work automation or finding economies of scale with groups of many workers is we look at the software development lifecycle. We take the SDLC, and imagine a matrix where the columns are phases overtime of a software project. You have pre-sale, kickoff, early, middle and late development milestones, handoff and maintenance. You could imagine each one of those is a phase. Then rows would be roles in the software development process. Development, design, project management, product management, QA, etc.

Each cell in that matrix is the work that a certain person does at a certain period of time in the project. For early milestones development, you could imagine a product that gives engineers a template so they can start working more quickly. That tool or that technology feature then augments that person at that phase, and that saves us time, saves us money. The client gets a better result. They get it faster.

As much as we can, we look at patterns in the data, inside of certain roles, inside of certain phases. One economy of scale when you have a lot of project managers could be that you've kind of seen the movie already, so to speak, when a project comes in. You know, "Okay. We're getting to this critical point — We're building a marketplace. There's this critical point we're

getting to in the projects that we see in all marketplaces. Here's what we've done at that critical decision point in other projects. This issue the customer had, here's how we fixed it last time."

We're building right now sort of a just in time AI assistant that looks at data from past projects and can actually connect A PM with other PMs that have had to go through a similar challenge. With development, we've reused some code across projects. We actually — In many cases, we license code to customers instead of sell it. We're able to then sort of open source that internally and use that across other projects if we can package it the right way.

We talked earlier about how the more projects we work on, the more we're able to better anticipate cost and timelines for future projects. We have another tool called the supervisor, which we use internally, and this is overseeing all the projects we're doing. We're doing about 150 projects right now. Average size; something like $100,000 to $200,000. We're able to look at all those projects, watch patterns. Like let's say there's a milestone due in three days. The developer hasn't checked code into GitHub in 5 days and maybe the developer and the PM haven't talked over Slack in 7 days. It's very likely they will miss this milestone, even though we haven't missed it yet.

You can sort of start to predict the future when you've seen enough patterns and you can catch bugs, you can catch lateness before they happen. Some of these technologies are actually fairly mature and used internally. Some of these technologies are fairly early. Collectively, we call these technologies that Gigster platform, and they seek to use data to find economies of scale across roles and across phases of the SDLC, if that makes sense.

[SPONSOR MESSAGE]

[0:32:42.9] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

[] JM:

[] RD:

recently prompted software about that they don't have the resources to execute souls there is always some new executive houses noonish and it's the most important thing in the world it's more southern executives and on and that there could be political battle internally resources are society we come in and see all be any political battles this will will give anyone technical resources and that one another example is the product before IBM completed 2 1/2 weeks and they were equipped to do that internally we basically came in and delivered a and we we please follow up really fast and seem like an AWS is within within days got back on you and the third thing we see is talent that they just don't house talking to companies as a German insurance company will only aunts and they have a ton of engineers internally but yet there are certain types of talent they don't house it might not have mission morning will come in and give you all engineers like that you know what I would say I would say it's the bandwidth or access to resources they don't have which is the case with the you would be the case law and work them in the case of the lumbar there is a narrative that every company is becoming a software company so if you're an insurance company you're actually a software company that happens to sell insurance I'm wondering to what degree you think that is the case or should an insurance company that has increasing software needs should they be scaling up there outsourcing for example and just focus entirely on their core competency of actuarial tables or probability of car accidents or something like that yeah this this is something we thought was a great way to plan, and questioned by the Mauer numbers you're somebody that Starbucks Starbucks is new CEO is telling people Starbucks is a software company which is just interesting because I read another their customers think of it that way you know what they do have software on our supply chain logistics customer loyalty through publications etc. so no I think you call a lot of company

software companies I fear my angel investing one thing that I look at companies is obviously a look at whether or not technology is the core part thing they have to do to win and if it's not I have no business investing because I don't know the types of business so I think if you call something next, he calls out then it has to be the case that technology is the core thing the when and I think some businesses that are consult software companies you may be maybe that's not actually true maybe the executive team at that moment time is excited about the results for the rallying cry to get the company more work going down the but I do think many companies are software companies any company that out some data for example insurance company is a technology company because technology data and their competitors more technology than they do and not wrong then they will you have companies that are really more about (something like 90 you know the credit Islamicist lighting but also their brand is so I think it really depends on the industry by you do do we see I think you see a lot more companies these days he's these days that are technology companies that it removes really every business in the world their own custom technology so I've done I've outsourced a lot of different projects that I've built because sometimes I like to just architect out an idea or do some wireframes and I don't want to write the MVP at this when outsource it I think the needs of an enterprise there but that's it's often the same thing they've got a new projects they want to spend up and they outsource it to to something like Dexter and then may have an MVP that they can work with and they can hand it off to somebody at the team there within their company that can manage it going forward and I do want to talk about that with you in a moment with her with the process that you know if a large enterprises getting something built how they work with with you with geeks are specifically up but just like talk a little bit more about outsourcing fundamentally in and how it might look in the future outsourcing is obviously getting easier it's getting good something has changed relative to how outsourcing worked and it your 20 years ago where it has become quiet and the ceiling value proposition and you know you know more about the fundamental economics than I do so I'm really not sure but could you imagine a world where the insurance company just says you know what our competitive advantage is where really good it doing sales and marketing and getting customers to come in through the door and and we just conflict due date it we get data and maybe we get most the out but most of our applications are outsourced me we have a couple engineers who worked internally to serve manage projects of their more project managers but they outsource all of their engineering work can you imagine a world where where that might happen were it's actually more economical rather than if you're if you're Allstate trying to higher software engineers and figuring at figuring out a an equity compensation

package it's really appealing to them you just do all of your work there outsourcing imagine that kind of world 100% and you your touch and I your previous question about going back and I really trusted is one of the reasons we started Dexter Leslie you know the vision for Ross is to be a central engineered for every business in the world: the department to and not exactly the idea that technology is essential for business yet most businesses don't really have easy access to it and I think it's the case that the great software developers are far less expensive to business than no not so great software developers you know the same total cost of ownership sense that we talked about before you we all talked about tennis developers and heard that idea what Tenex developer doesn't cost the company 10 times more than a one external 60 point as a business like an insurance company right is a competency how do you attract a bunch of Tenex people I don't want work for so I think what the future might look like there is one central are several central entities that do most of the and the great talent is co-located in those central entities and companies have a thin layer technologists but they mostly focus on their brand their core value proposition may be maybe the insurance company really has one out differences every other insurance company sort of like you know this secret sauce a special ingredient the restaurant might have its entire businesses they stop but after that they don't really need to worry them you don't bother consult details I think were already seeing that were sitting on a lot of our customers even before we got there at other companies that though most of the technology I can name specific names but a lot of our customers don't really have much technology sort of a skeleton crew internally on you it makes sense due to market trends in the the best engineers want to work with good people on interesting problems and there just aren't that many some of these companies so I think you're seeing the best engineers go-go frameworks are seeing them go join organizations like Google and Facebook there's that point the rich get richer dynamic which which really hurts if you don't have engineers already so building a culture understanding how to work with engineers simply won't be a competency a lot this will give a little more into that later on but was give people understanding for how gigs stir works describe the process of an enterprise coming to work with geeks to we typically you know the lease only resets us what we try to do very quickly is understand that a high level what the problem is they're trying to solve with technology rather than get down get down the rabbit hole of a conversation about a specific mobile lot website they want to build then we go from there sometimes will do his will will do if the requirements are clear-cut upfront will do something engagement were will will, and will help them help them articulate what they wanted bills will create a suspect document and will give that they can implement doctrinally to give it to

development solution where they can handle right back to us which case little baby is they come to us for the specification already made that we can go back so the where process for development works is we need to have clearcuts that we don't you have to work everything we do is size closer to water all so it's it's fixed-price scope defined we allow for some reasonable flexibility during the process but we do require an up front will find so then we manage the process through series of milestones the client on US developers are paid by the hour there paid by the milestone of the we and we are by the customer also milestone so none of it is hourly basis that's not a staff augmentation model we will give you individual developers we only give you an entire team and the reason we do that is understand what you want we have full control over the see how the elections operate we can ask the optimizer happens under the and the success that Huber don't not a marketplace for you to meet specific drivers and hire them? Anyhow finds you the best person maybe someday I will give you a soft running car or it'll just tell you it'll do the most efficient thing to get to your destination we think of putting a software team together and deliver his consultants digital ocean space simple objects storage with the beautiful user interface you need an easy way to host objects like images and videos your users need to upload objects like PDFs and music files digital ocean built spaces because every application uses objects storage spaces simplifies objects storage with automatic scalability reliability and the low cost for the user interface takes over the top I built a a lot of web applications and I always use some kind of objects storage the other objects storage dashboards that I've used are confusing their painful and they feel like they were built 10 years ago digital ocean spaces is modern objects storage with a Modern UI that you will love to use it's like the UI for dropbox but with the pricing of a raw objects storage I also use it like a consumer product to try digital ocean spaces goaded D.O.co/SE daily and get two months of spaces plus a $10 credit use on any other digital ocean products give us credit even if you have been with digital ocean for a while you could spend it on spaces or you can spend on anything else in digital ocean and it's a nice added bonus just for trying out spaces the pricing is simple five dollars per month which includes 250 GB of storage and 1 TB of outbound bandwidth there is no cost per request an additional storage is priced at the lowest rate available justice sent per gigabyte transferred and two cents per gigabyte stored there won't be any surprises on your bill digital ocean simplifies the cloud they look for every opportunity to remove friction from a developers experience I'm already using digital ocean spaces to host music and video files for a product that I'm building and I love it I think you will to check it out D.O.co/SED will he get that free $10 credit in addition to two months of spaces for free that's D.O.co milestone model at the

geek keeps incentives aligned you don't have engineers purposefully taking a long time to finish a milestone that is finished the milestone as quickly as they can and then they get your money based on that so how do you figure out how to agree on a price up front and how do you prevent scope creep because a lot of people that you're probably interacting with their use to agile webbing I like the you the you own that the term waterfall I also like the you own the term outsourcing I don't see either of these things is as bad words there kind of the been the of maligned due to the disparate experiences that people of bad but told me more about how you figure out what the price is going to be when a company comes you in there like a we wanted to project ask your kids gonna cost acts and here's the milestones were to break it down into and how do you prevent the scope creep so so this is one of the areas that are core technology comes in waves it in a sense please mount the software genome so we understand what a new project comes in we target with different details we understand what features is made up of us will project has various mated variables like what technologies he uses when there's a rush job or not what what platforms is adopted in the existing customer technology and that it has a list of features you know for the upload settings page landings page of feed comments likes to eat each of these could be different features in an application we tag a new project that comes in with these features what that allows us to do is find a similar past projects and we have what we've done thousands of projects for clients so we can actually find very similar projects we can look at data on how this went this data comes in from of from Gerard of control oh from slack from you know whether it's communication project management or source check in we aggregate all this data we can get an understanding of how the project went use that's predict the optimal timing price for future projects and we got to a point with anorexia very accurate so we know where we we retain our margin probably 9899% of the time of projects which which is which is pretty stunning but and also something that structurally also model how much do you talk about that margin how much the developers are getting versus how much geeks to captures is that confidential yes it is it's also really just a pencil projects the margins lonesome project because we need to redo work or we have a lot of internal company resources like customer support that has to have always supported our freelancer base you know and in other cases it ends up being higher wearable to reuse code assets across projects since it's really highly project appended but the reality is what we do for freelancers we take on the risk so they can just sensing right coda get paid we try to make of the easiest possible interesting model for how the client is interacting with the developers that are writing the project so let's say I am a company a big enterprise and I come to get exterior because I need this machine learning

application built eyesalve got all this customer data and I want a machine learning algorithm that's going to predict whether this customer is the return or not hears, the data here's what I'm looking for and the customer sorry the the enterprise that you're dealing with will actually interface with a project manager and the project manager will manage the engineers on the dexter side so the engineers actually don't ever interact with the enterprise that's coming to get exterior as a customer is accurate that's right most of it in the most most cases you know our engineers will reasonably work with us as they don't want to interface with clients they could join another freelance network and they can work directly for clients and staff augmentation type arrangement but engineers like us because we have that right somewhere we protect them we make sure they're paid on time the project manager is is typically freelancer as well although we do also have internal resources that oversee the whole project so really it's an entire full functional freelance team where all the development resources are you know, be directed from the customer with with that project manager that's always been a model and it's it's it's a it's a bit essentials you know to to have that that kind of quality control to manage service later to make sure they were able to margin given that were guaranteed a fixed price which is why is it so important to tuning the marginal corners because if if the client was working directly with the goal we might see is the client randomizing the developer scope creep in the project over to do extra work that wasn't part of the original project plan great developers don't want you also have to be project management and you know the future creep with clients as her as the building is the project so project managers are this line of defense in a way where the client comes in tries to change the requirements for the client you know doesn't like to deliverable the project manager can no cutting solely the development from that with you know we are project managers have to push back sometimes sometimes it's the opposite sometimes we try to overdeliver for the client but it's it's nice to have that layering to make sure that someone's watching me solutions so the project managers are product product managers at gig stern are they managing multiple projects concurrently in some cases they are we have project managers that manage up to I would say he projects it wants something in Iran to manage one or two projects and others that are between projects maybe there there full-time gig has a busy start up or a small business or a family issue on the side so we we we don't mandate that they work any amounts long as they're able to do good job can use many projects what sue us one thing I was I was interesting in is how the how gig stir its self skills and is the number of project scale so if you you just have the income if you would be a linear increase in the number of projects that were on gig stir I guess the the number of project managers are product managers that you

would add would scale sub- linearly because that every additional product project manager can probably manage multiple projects you probably get some economies of scale there more about the economies of scale that you get as you get more clients coming in the door slowly and I think it's a summer project management others are around developments and quality assurance and other areas the way we think of so HE model lances directly the way we think about kind of work automation all say or know this future work concept of work automation fine economies of scale so many workers is we look at the software lifecycle take the SDLC and imagine imagine a matrix where the columns are phases over time of the software project so you have presale kickoff early middle and late development milestones and often maintenance those imagine each one of those is a phase and then Rose would be roles in the soft adult process so developments design and project management product management way on each cell matrix is the work that a certain person does a certain period of time the project right so for for early milestones development you can imagine a product that is engineers a template so they can start working more so that that tool back that that technology feature that augments that person at that phase and that that saves us time saves us money the client gets better resulting in faster so as much as we can we look at patterns in the data you know inside of certain roles such as certain phases so 11 economy of scale when you have a lot of project managers could be that you you kind of seen the movie already so to speak when you when project comes in and you know okay were getting to this critical point you were pulling a marketplace this is critical for getting to the project that you did we see in all marketplaces and it's here's what we've done at that critical decision point and other projects know this is for the customer writers are research the last time so we're building right now sort of it just in time AI system that looks at data from past projects in connection connect a PM with other PMs that had to go through a similar similar challenge developments we we use some cross projects so we actually in many cases license code to customers instead of solid so you were able to then sort of open source that internally and use that across other other other projects that if we can package it the right way and we talked earlier about how the more projects we work on the more were able to better anticipate costs and time once for future projects we have another tool called the supervisor which we use internally and this is overseeing all the projects are doing so are doing about 150 projects right now your average size something like one to $2000 were able to look at old projects watch patterns like what Sid is a milestone due in three days the developer hasn't checked coda to get up in five days and maybe the developer in the PM how to talk to the socket 7 days it's very likely that he will miss this milestone you know we have invested so you

conservative start predict the future when you seen enough patterns and you can catch bugs you can catch lateness before they happen some of these technologies are actually fairly mature and in use internally some of these technologies are fairly early but collectively we can always technologies the extra platform and they seek to use data to find economies of scale across roles and across phases of the estimate is wearing

Spring Framework gives developers an environment for building cloud native projects. On December 4th through 7th, SpringOne platform is coming to San Francisco. SpringOne platform is a conference where developers congregate to explore the latest technologies in the Spring ecosystem and beyond. Speakers at SpringOne platform include Eric Brewer, who created the CAP Theorem; Vaughn Vernon, who writes extensively about domain-driven design; and many thought leaders in the Spring ecosystem.

SpringOne platform is the premier conference for those who build, deploy and run cloud native software. Software Engineering Daily listeners can sign up with the discount code sedaily100 and receive $100 off of a SpringOne Platform Conference pass while also supporting Software Engineering Daily.

I will also be at SpringOne reporting on developments in the cloud native ecosystem. I would love to see you there and have a discussion with you. Join me December 4th through 7th at the SpringOne Platform Conference and use discount code sedaily100 for a hundred dollars off of your conference pass. That's sedaily100, all one word for the promo code

Thanks to Pivotal for organizing SpringOne Platform and for sponsoring Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:34:15.1] JM: You have worked as an engineer at a large corporation. You worked at National Instruments back in the day. I've also worked at a large enterprise. I have trouble imagining myself ever going back, partially because I like working for myself, but also partially because I like the freedom of working outside of an office, and I think if more people tried, that they would

find how preferable it is in many context, unless you're building your own company, you're building a small company. Building Gigster I'm sure is exciting going into the office.

The reason I'm mentioning it is because I'd like to know about the psychology of the type of people who come on to the Gigster platform to make their living. Can you describe how those people contrast with somebody who might go to a large corporation, like a Google or a Facebook, or National Instruments?

[0:35:21.7] RD: Certainly. We actually designed Gigster from the developers perspective first and foremost. We're basically a marketplace, right? Customers come to us and they look for engineering resources, and whenever you're designing a marketplace, you need to look at where you have the largest constraint, whether it's supplier demand, and you focus on that constraint, you solve for that and everything else works itself out.

I think it's sort of a truism that if we had all the best engineers in the world on Gigster, that we'd have a lot of business, right? We ask ourselves, "Okay. How do we solve that problem? Because, clearly, that's the most important problem.

Granted we're product for customers, and the product is they get great software built for them, but we're more so a product for developers and we think very carefully about how we can design what we call a great career experience on Gigster. From the get go, my cofounder and I asked ourselves the question, "What would it take to get us to freelance? Why don't we freelance on ODesk, Elance? Why don't we do it personally? Why aren't we on TopTal or something else?" The answer was a couple of things. It was some small finite number of issues with all of these freelance solutions, that if solved, we'd be delighted to work, that we'd be delighted to freelance.

One is developers don't want to market themselves. They don't want to do sales. They just want to write code. Two; they want to work with interesting people on interesting problems. They don't often want to engage directly with the client. They like to get specifications in consistent clear format, versus every single client you freelance with has an entirely different development process, entirely different set of documents.

We created an experience that kind normalized all of that for them and allowed them to earn a Silicon Valley wage from wherever they work in the world with interesting people, cool projects, reliable payment, no marketing, all that stuff. We solved all those problems and we've attracted some incredible people. We're in 55 countries. It's 70% U.S., but 15% Western Europe, 15% other countries. We've attracted people who kind of in the same sense that — In the same sense that Google has this employment brand where they're the best place to be a full-time engineer, or one of the best. We've tried to do that for freelancing. Maybe attracted some incredible folks. A number of people from the recent MIT class found out about Gigster, because a student was making a lot of money in the platform and they just graduated and became Gisters.

We have a lot of people that moonlight from jobs at SpaceX, Google, Tesla, Facebook, eBay, you name it, top tech company. We have people from that tech company moonlighting. We have a lot of startup founders that are looking to kind of pay the bills on the side. Maybe this startup isn't doing that well. They're running low on funding. We're probably funding more startups right now than at Silicon Valley VC. Then you have people that just want a different career. Maybe they live in the Midwest. They have a family to take care of. They want to code on the beach in Thailand, whatever it is. We're making that possible for people.

I was just as surprised as other people happen by this. I didn't know how many people actually wanted to freelance. I didn't realize how much latent demand there was for freelancing, but we started Gigster and people have come out of the woodwork to participate. I think there's just a lot of people that want to work in a different way.

[0:38:40.1] JM: What's the hardest aspect of scaling the Gigster market?

[0:38:44.9] RD: Scaling the marketplace?

[0:38:46.0] JM: Yeah.

[0:38:46.6] RD: I would say it's a few things. One is project management. Finding great project managers is actually really hard. Whereas we have technology right now that optimizes and augments human project managers, we're by no means replacing human project managers.

There's a tone of human coordination. It's been very hard to find people that can help a project run smoothly. We've seen seem projects that are going terribly wrong. They're buggy, they're three months late, and the customer is happy as a clam, because they love their project manager, because they get on the phone and crack jokes with them. There's a personal relationship. There's a very human side to that, which is important.

On the engineering side, we heaven't had as much trouble attracting great engineers. I think we're highly selective, something like a 1% accept rate if you look at the overall pool that we consider. Once people come in, as long as they get work, they're fairly happy and haven't had as much trouble there. Design hasn't been as difficult. Product management is another bit of a challenge. We've actually mired the business. You said at the beginning of the call that we serve enterprise. That's relatively new. When we were founded three years ago, we were serving SMBs. We're doing 10, 15, $20,000 mobile projects. Now, our sweet spot is something between 100,000 and several million dollars.

We're working with different clients that are more discerning and the people that are client-facing need to be more experienced and more polished. We're going through a bit of a reboot in some of the roles where we have to recruit different types of people, but I wouldn't say that's challenge so much as it's just something we have to work through.

[0:40:30.5] JM: Was the change from the 15 to 20K clients, to the 200, 250K counts, was that deliberate or was it — What caused that change?

[0:40:44.1] RD: In two parts it was deliberate in a away. We recognized that that's for the larger market for software is and we wanted to have a better avenue to grow our business, so we went after where customers were spending more on technology, which is enterprise, which is midmarket. It's not as much SMBs. If you think of us like a SaSS business — SaSS is great because you're in a recurring revenue from a SaSS product. You can predict the tenure of a customer. You can predict the lifetime value. With services, it's different. You do have services relationships that have lasted for decades. I'm sure there's people that have been using IBM consulting group for 20 years and they've been consistent clients, but it's not as predictable as the ARR that you get from a SaSS business.

We found with SMBs that a lot of them didn't repeat, because maybe they ran out of money, they went out of business or they don't have as much budget for technology. Part of it is we did go after bigger budgets. The other part is we don't feel good not delivering a product that makes our customers really happy.

What we saw some SMBs and midmarket customers doing was optimizing for kind of a bargain-basement deliverable that didn't move their business forward as much as they could have. Whereas if customers that have budgets that's in the six-figure range are able to get something that's a much higher quality, much more durable product. We found ourselves wanting to spend more. We actually went into our margin a lot on customer projects. We use to typically make almost nothing on client projects, because we would over-deliver and just try to give them a better solution until we've realized we simply had to raise prices in order to deliver a product that we felt proud of. That was part of it. Then as we raised prices to deliver better products, that pushed us into higher markets. It's sort of part organic, part strategic.

[0:42:34.9] JM: That sounds like a really interesting and kind difficult process to manage. If you were jacking up the quality and then that ate away at your margins for the 15 to 20K deals and you said, "Well, gosh! We can't make money on these. Let's raise the price," but then maybe you — There's an uncomfortable chasm between the 15 to 20K deals and the hundred-K plus deals. I'm very curious about like the conversations and the kind of like a spreadsheet analysis that led to — We really need to like cross to — We need to go completely into enterprise. We need to burn the ships, abandon the 15 to 20K contracts. Explain a little bit more what happened behind-the-scenes.

[0:43:27.4] RD: When we raised our series-A, I think this would have been November, December 2015, we announced it and we had $30 million of business come in a single week. We didn't close all of that. Thank God, because as we would've killed ourselves trying to deliver it.

[0:43:41.8] JM: These were — Sorry, 15 to 20K contracts.

[0:43:44.5] RD: I mean it was everything, from 20K to 200K. We weren't seeing anything really much higher than that. It was 5 to 6 figure range, 5 to low six-figure range contracts and just a

ton of them from all quarters. Everybody saw the news and they're like, "Oh! I can push a button and hire a development team? Great!"

We had a ton of people sign up and we took on a bunch of customers and our delivery operation kind of grind to a halt. We weren't ready for that much demand. We didn't have the supply side quite as shored up as it could have been. We ended up going to the red on a ton of different projects, because the invariant for us is we've got to do a job. We guarantee quality. We guarantee price. Whether we go over or not, we'll never charge the customer more.

It was a stressful time, and we started inching prices up sales. We'd tell them, "Hey! We've seen that marketplace projects typically go wrong in these few areas. You need to charge an extra 10K, an extra 20K next time," and they said, "Okay," and they'd say that to customers. Then we noticed something else that would go wrong, and then we'd inch the price up a little bit more.

What sales started having to do, because a lot of our customer base back then was still those SMBs, but the prices were starting to rise. What they have to do is they had to de-scope with clients. Clients would say, "Look. I've only got 50K and I need this tinder for dolphins app," or whatever, "How do I get that done 50K?" We'd say, "Well, are you willing to sacrifice this feature and this feature and this feature?" We got into all these dialogues of people about how to de-scope so that what we actually delivered would be high-quality. That worked for a while, but in the process we had to start looking for new types of customers and that's where some of our investors helped. Andreessen Horowitz was incredibly helpful at helping us boot up a bit of an enterprise and midmarket business.

We worked our personal contacts and connections from previous large companies. We had to reinvent the company in a lot of ways. As you said, it was definitely an uncomfortable time, but now we're in a much better place, where we take on a project, we know we can do a good job.

[0:45:52.6] JM: Have you had to build an outbound sales model?

[0:45:55.5] RD: We have. Early days, everything — We thought it would be self-service. My background is consumer internet. I used to work at Zinga. When I was managing games at Zynga, I founded a game called Mafia Wars. We would get a half 1 million new users per day for

free, where just sit there and 500,000 new people a day would come in and try the game. Many of them with sticks and many of those would spend money. It was amazing. It was the gold rush. It was a great business to be in, and consumer has that profile.

When I found Gigster, we this idea of putting a software development team in everyone's pocket, and we embraced that to the extent that we didn't want a sales team. We actually wanted the salespeople to be freelancers. I designed and built a freelance sales CRM where anybody could sign up as a salesperson, go through a brief training course, and then they were entered into this sort of round robin queue, like an auto dealership. New customer comes in the door, you stick a car salesman on them and then that person owns the customer for the duration of the relationship.

We would take the three-minute salespeople, assign them, and it was all data-driven. There was no sales management. We would look at how many seconds does it take you to reply to customer chats. What percentage of customers do you deliver a proposal to? What percentage sign the proposal, and then what percentage of them do repeat business and how many of those projects go well? The idea was we could throw in a thousand salespeople who could work from wherever they want, whenever they want. We could scale to an unlimited number of inbound leads.

Unfortunately, those salespeople didn't always act in the customer's best interest. We ended up having to hire salespeople in-house and then we dismantled the entire freelance base sales system. We should patched it up and sold it, because I like the idea. We originally thought of this thing where like everybody's mom and brothers coming in and building a little app. We couldn't afford sales or our margins would go to zero. Now, we have to go. Like you said, we have to go outbound. We have to go direct enterprise. We have to have a direct sales team and a direct sales model. That's how enterprise works. You want to come in and help them solve a problem. You need to send in someone who understands partnerships who can actually articulate the proposal and get it done.

[0:48:09.3] JM: And find the right person to propose to. Who are you selling to at a big enterprise? Are you going to the VP or the CTO and just say, "Hey, do you have a 200 K to throw at a project?" How do you find the right person with the project?

[0:48:26.8] RD: We either come in — Sometimes we come and top-down as soon as we come and bottom-up. Sometimes I know where I can get engineers, even the CEO of Fortune 500, or a CMO, a CIO, a CTO. Usually, it's one of those executives. Sometimes I come in through board members. Marc Benioff is one of our investors. We've got a lot of investors. We have sports teams owners as Gigster investers. We're able to link those, typically to those executives, and then they'll say, "Oh, yeah! I think we have a couple of projects. Let me kick you to the VP of North America innovation. Talk to Sarah."

We come in to the top, we'll then get kicked to somebody who has an actual need and then we talk to that person what their actual need and that decision maker usually has budget, and then we can work with them. There's an acronym and sales called BAND, which it stands for budget, authority, need and timeline. If we meet somebody who has the timeline, they're ready to start, get a real need. They have the authority to actually pull the trigger and the budget to do, that's ideal.

That doesn't always have to be the same person. Sometimes multiple people constitute that BAND that you need to have. Yeah, our sales team essentially seeks that out and once we find a specific project and all the right decision-makers around it, I simply deliver proposal and get this on.

[0:49:49.2] JM: Alright, I know our time is short, but let's close off. Do you have any insights about large enterprises? How large enterprises function these days from your experience selling Gigster engagements with them?

[0:50:07.2] RD: Yeah, I'll share a couple. Number one; most large enterprises have their own internal technology teams. They've built their own many agencies internally, and oftentimes you're competing against — If you're an outsourcer, an agency or competing against this internal agency they have. If they don't have an internal financial agency, they certainly have an external agency that they've worked with for years. Often times the right path in is to partner with one of these agencies and perhaps had some skill that they don't so that you both together can create more business.

Enterprises are fascinating with new the technology. Everybody's interested in AI, machine learning. They see it as an incredibly disruptive technology. The way I would phrase it, and many of our customers would agree, is that AI and ML, this technology represents a new trend on the same scale as the internet in a sense that all business software will be rewritten to accommodate AI and ML just the way it was rewritten to accommodate the internet.

Most of these companies recognize that almost all of their software will be rewritten in the next 5 to 10 years and they're looking at the right way to do it, “Do I hire? Do I outsource? Do I buy an off-the-shelf product?” I’d say there are a lot more questions than answers. There's a lot of interest, a lot of passion for technology. It’s something that's universal.

I’m based in Silicon Valley and I see a lot of innovation here. I see a lot of great startups, but what I didn't realize until I got into the enterprise market, is just how innovative enterprises are. One of my favorite parts of my job is getting to meet at executives within enterprises who are innovating. They have cool ideas and they’re like, “Hey, I work at Microsoft, or Nike, or MasterCard, or Starbucks and we have this incredible asset, and I consider myself an entrepreneur,” in that case they’re an intrapreneur, “and I’m going to take the resources behind me and I’m going to develop an incredible new product or feature or spinoff a new business.”

I love finding the executives within enterprise that are the innovators, and they’re there. We might not think that Silicon Valley might have a bias against enterprise, but they're incredibly innovative, incredibly brilliant executives that are starting new lines of business. I love meeting those people and helping be their technology department essentially. I was surprised to see just how much innovation is happening. Those are, let’s say, a couple of insights I think.

[0:52:37.6] JM: Thanks, Roger. This has been great talking to you, and I want to thank you again for coming on Software Engineering Daily.

[0:52:44.2] RD: Okay. Certainly. Thank you.

[END OF INTERVIEW]

[0:52:47.7] 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. That's 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]