EPISODE 1524

[00:00:01] J: Welcome, Jason. It's nice to have you on Software Engineering Daily.

[00:00:06] JDS: Hi, Justin. How you doing?

[00:00:08] J: Good. Good. It's good to connect. I thought maybe what we could do is just start with a quick introduction of you and your background, and then get started talking a little bit about Codat, and open finance and data.

[00:00:21] JDS: Yeah. My name Jason. I'm Head of Engineering at Codat. We're a business that does data aggregation for financial information. I've been writing software and building things for about 15 years. I started in embedded systems and things like that and moved up through this OSI stack, I guess, until now. I'm doing big cloud deployed web application type things. Yeah, that's me.

[00:00:51] J: Great. Well, I'm really glad to have you on the show. Because, as you know, data is a topic that's near and dear to my heart. There's a lot of new capabilities being unlocked right now by folks like Codat, other new thinking, cloud architecture. One of the big Ideas behind that I wanted to start with is you hear a lot about open finance, open banking. I was hoping we could just start with your definition. How do you think about the notion of open finance?

[00:01:21] JDS: Yeah, I think open finance is sort of a set of ideas that I think a lot of different services are kind of coalescing around. And it's happening I think in little bits and pieces in lots of different corners of the world as slowly we get used to this kind of API software, internet-driven sense of connected systems. And sort of giving the hand, the reins back to the data owners in how they choose who they share information about themselves, with and why. And nearly, always, it's data that's going to be super useful to automating processes that people have been doing maybe the long way for a while.

And I think in open finance, the biggest bonus is going to small medium-sized businesses where business owners who have got usually a lot of jobs and a lot of hats to wear. They spend a lot of time doing financial administration for business and an open finance a lot of times is about kind of reducing the friction.

And so, sometimes it's not obvious. Sometimes you don't see it necessarily. It's not on the face of things. It's happening under the hood and it's just this sense of reduced friction and this expectation of being able to share your information where it needs to be in order to reduce that kind of administrative burden.

[00:02:43] J: Interesting. You get an improved customer experience. You get to reduce risks. All the friction that we see today is largely because there's so many silos of data, right? And so, organizations are required to tick and tie all of them together in order to laboriously get to acquiring a customer in real time or providing a customer experience or shopping experience in real time.

The vision is to have this integrated financial experience based on an integrated view of the data. That sounds easy to – it's easy to say and hard to do. And I want to talk much more about that. But today, what do these small and medium-sized businesses get as they move towards open finance? What are some of the specific benefits that they see? Certainly, we say frictionless. But day-to-day, what do I get as like a benefit of this integrated data view?

[00:03:48] JDS: Yeah. I think a lot of this has to do with choice. I think there are a lot of different systems that business as a consumer can opt into to do some work for them. And so, there's a lot to be gained from picking tools. They're going to solve problems that you have. And the friction then is like, "Well, the more choice I have in terms of the systems I use, maybe the more the administrative cost is of being able to move data between those systems."

I think as these become more connected, it opens up choice. Choice about who you need to share and when you share certain information with services to gain some some aspect of some product offering that they have. And also gives you more choice to pick and choose what systems and services you think are going to serve the needs of your business best.

And it's going to help them run their internal operations, those things, and save them time. Because you're picking that tool that's going to be the best for the job probably to save you time, or lower risk, or whatever. And the more and the better that they talk to each other, again, you're reducing that friction.

[00:04:55] J: These are some really difficult areas of financial management for small and medium-sized businesses, for everyone really, right? I think what you're kind of putting your finger on is like there's integration. Getting all the data sources together. There's this reconciliation of what matters and what doesn't in a given situation. And then just automating the risk activities, right? There's probably more to it than that. But those are a couple of things that jump to mind as you're talking. And can you give me a sense of how small and medium-sized businesses are doing these activities today?

[00:05:37] JDS: Yeah. I guess on the integration part, if those two systems that you might be using are not already talking to each other, then there's sort of the competitor to that in a not sort of open finance or connected world is you're printing off a PDF, right? You're sending your bank statements hopefully that you've managed to be able to get off your online banking system if your bank supports that. Or you've got some stack of paper that you're sending to someone and you're maybe using the same thing for reconciliation, right? I'm going through my bank statements. I'm going through my ERP system or something like that.

If some business who cares about my operations of the business wants to get an understanding of the operations, they might be like, "Well, I'm going to trust a bank statement." I think all of these things that isn't necessarily a competitor directly to the non-integrated world for that portion.

And really, for someone on the other side, all of those different – and we talked about the choice before. That means there's a huge amount of fragmentation start to appear. Either there's every single bank in the world or every single ERP system. Which means that the consumer of that information on the other side that's going to be offering your product needs to have some way of kind of understanding that either through some standardization or some aggregated way of being able to consume it. Yeah, I mean, that's hugely important. The reconciliation point, I guess? Is that the next –

[00:07:16] J: We'll talk more about that. I want to talk about sort of about like integration. Like, how do you get the data? And then how are you standardizing it in a way that it is apples to apples comparison? Something that you can use all together for multiple sources for one decision. And then I want to talk about sort of automating that process and making it faster and frictionless, right? Those are, to me, the three big parts of the narrative.

Before we entirely engage in that, I thought maybe you could just – can we just walk through a use case to just understand the sort of before and after picture? And so, tell me if I'm right or wrong. Because I think customer acquisition is one use case I've heard about. Is that a good one? Should we walk through that for a second?

Today, I feel like, if I was a merchant or trying to offer this service to my customers, it's lengthy. I have to send information into the acquirer. There's a wait period. Maybe I would end up sending a letter or, a couple days later, an email, like, "Hey, we've done all that you wanted us to do. And now you're an authorized user, authorized to you know spend this much in my site or my world."

For neobanks, fintechs, what they want to be able to do is immediately look at multiple sources of information, evaluate that risk and immediately acquire that customer. Is that kind of the before and after picture?

[00:08:45] JDS: Yeah. Yeah. That kind of conversion journey I guess is always going to be better informed. The quicker you can get that information that's going to understand whether you can qualify that individual that's trying to onboard on your service, the better. And by getting a view of that information to make those sorts of risk decisions or make that kind of decision quickly does require some sense of standardization. I think that's what we're kind of alluding to. And that's obviously one of the things that Codat provides.

The onboarding piece is also often – there's a funnel that's going to drive that particular engagement to even the point of data sharing. And I think that's another thing that Codat feels we're pretty well-placed for, which is a link product that we have, which is really where that journey starts, which is something that we've built that can be embedded directly into someone's onboarding offering. If they've got some sort of sign up flow or something like that, they know that you can embed Codat in there to say, "Hey, look, you're going to get a much better service.

It's going to be much easier to administer. And you're going to get a risk decision much quicker if you share financial information either from your ERP, or from your point of sale system, or your bank statements, or a combination of those."

And we've got a number of products. We've done that with a lot of different accounts, different clients, customers, for a lot of different systems and a lot of different data sources. And we've just released actually a set of documentation specifically geared around, "Hey, we've run some numbers. And we think we know where we can help you guide to better conversion rates on that funnel for data sharing." That's one place where it's not like onboarding and those kind of – that conversion funnel is necessarily some particular skill set. I think it's more by observation. And then supplying tools to help other people do it better.

Because, actually, that journey and understanding the needs of each relationship or engagement that you have where you should be directing them is potentially – it's going to be a very important for that –

[00:11:00] J: That's interesting. Yeah, yeah. If you have like a multiplicity of sources and you're able to look at them in some integrated, standard way, then what you can do is make observations automatically instead of declaring a hypothesis and searching. Like, having this declared human-driven process, you would have more of a machine-driven process. Yeah, yeah, yeah .

Let's talk a little bit about – So, we've talked a little bit about the promise. All the data is integrated and it's all apples to apples, which that's easy. We'll talk about that later. We'll get there.

We'd want a solution that does so much of what you're saying, right? It makes it easy for me. I don't have to have my own software development group to connect to everything, share information, authorize it accurately, analyze it and then act on it, right? You are going to make that easy for me.

One of the things I'd love to get some help on, Jason, how to think about Codat? What are the key products? Right? And how should I think about the framework in which they're offered? How do you guys talk about it?

[00:12:11] JDS: Yeah. I guess, first and foremost, it's that we're building a product to help reduced friction for SMBs, small and medium-sized businesses, to share their information and with other organizations that they want to buy products and services from. And we add a layer of a number of products on top of that core concept in order to enable various use cases for companies that need that information.

I talked maybe before about this kind of link, this kind of standardized method of authorization that can really help with onboarding and converting people to sharing that data and understanding what data they can share. And then there are a couple of other sort of key product areas that sit on top of that. We have an assessed product, which is for digital lenders. That idea is it's kind of incorporating that data into whatever underwriting process that organization might have. And a sync product, things like our sync for commerce, which moves point of sale information into an accounting system. And sync for expenses, which is, again, moving kind of expense information into your accounting system to account for it. And that's like a great loop, right? You want to be taking all the information you have about the behavior of your business and centering it in a system that you're going to be using for reporting, and then allow you, as a small business, to kind of share that information in this standardized way in order to get better access to products and services. And that's good on the other side, for those that are using Codat, that they can build those sorts of products without necessarily having to worry about the difficulty of that data aggregation portion.

[00:14:12] J: I'm so glad you paused on this automated reconciliation component that's supported for COMP SYNC. It's so incredibly painful. If anybody has ever tried to do that, connect these systems and get some kind of even like baseline understanding of how payments are going in and out. What you owe? What people owe? It's still incredibly difficult.

[00:14:42] JDS: Yeah, in some instances, there's kind of a regulatory overhead about how that information needs to be presented. And that can come from the data, kind of the system that's providing the data about how they feel that should look in an accounting system. How you account for that?

Also, the ERP system itself that may be holding that accounting data might have stipulations and uncertain regulations around how they want that data from certain sources to look in their system, right? And normally, that's kind of going to be a manual process to your accountant or the business owner. Kind of sat down doing data input.

And, yeah, it's really difficult. And the same with the reconciliation for things like expenses. And I think we saw something with brex. It was really interesting. That was actually the volume of spend in a business in terms of how they spend in terms of expenses is something like eight times more when that business is connected to their accounting system. I mean, because the nature of the integration making that smooth, right? Makes that process easier.

[00:15:58] J: Right. And transparent. Yeah. No. That's the demo. If you haven't made it, I think that's the demo I would love to see, which is, "Hey, I'm a small business owner. Here's my website and my storefront. And here's my accounting software." Right? And just being able to immediately, in real-time, continue just keep that ledger rolling in a way that I can report on it. I can get the transparency. Take action. That's something small and medium businesses certainly haven't had access to. They're very underserved in that place. Aanyway, I know you love the product. I'm just saying it's a very – I just want to underscore for the audience, this is a very underserved group and it's very painful to do.

[00:16:40] JDS: Yeah. I mean, you can imagine, we've got a London office. Just outside our London office, there's a food market. And every time I go to buy something while I'm here for lunch. And you use a Zettel reader. I know that that transaction is going through that point of sale system. Now, quite often, the guy that's serving you the burrito, potentially the business owner, that transaction that's just going in through that Zettel reader. And it's going to be synced and reconciled in their accounting system by the end of the day. That means if that business that's selling burritos next week needs to look for some working capital in order to buy another hot plate because they're selling loads of burritos, I know that the behavior, like the reality of how that business is performing from the previous week is going to be there readily available for somebody that might be able to offer them some service to –

[00:17:36] J: Yeah, and that's the kind of actionable transparency. That is really powerful, if you will get into the technical problems. But in the promise of it, that's something small and mid-sized business haven't had, this idea of access to risk analytics, automated and risk assessments. And then this ability to sink into your ledger, right? And understand your supplies. All those things have just not been available. And then I kind of cut you off, because I think there's another part of the products you wanted to talk about. Or maybe I didn't cut you off.

[00:18:16] JDS: No. Yeah, that was the loop.

[00:18:21] JDS: Okay. Okay. Good. Good. You have to stop me. I'm an interrupter. But, all right. That is the promise we talked about, right? And those are the products. But I do want to kind of dig in and talk about the technical underpinnings of accomplishing, delivering those products and having them be adopted and work correctly. To me, let's talk about you'd have to have the integration. But you'd also have to have really powerful authorization, right?

[00:18:52] JDS: Yes. Yeah. Yeah, for sure.

[00:18:56] J: How are you thinking about – how do I authorize who on my small team can – Or authorize which data talks to which? Or which organization gets my data? Or doesn't?

[00:19:07] JDS: Yeah. We have this idea of standardized authorization on the basis of each kind of relationship has a connection to some data source. You can set up a product we called Link. You can set that up essentially to define your workflow for what sources of data you want in order to be able to serve that particular engagement, that particular relationship.

And that will take them through a guided experience. And essentially, the standardizing part of it is a bit that kind of workflow of getting this consent. Now, most of the sort of the systems that we're integrating with, we use some sort of like three-legged authorization, OAuth 1 or OAuth 2. Obviously, OAuth 2 is very popular, to guide those relationships through the workflow to get consent. We act on behalf of the individuals who use Codat. And of course, of course, there's the sensible security concerns over the kind of lifetime of tokens and things like that that we will manage.

[00:20:15] J: You're managing all that for me if I want you to. Is it possible for me to build my own and still use the rest of the product?

[00:20:23] JDS: Yeah. The way that most of those systems work is on the basis of some short time, some short running token. With something like OAuth 2, you want to be biasing towards the token that's on the wire, which is the one that has this this access is as short-lived as possible. So, that it's ever particularly compromised while it's out in the wild in this untrusted world of the internet, that if you've got the shortest possible time at which that you could be exposed by that.

So, we have to manage the lifetime of that. Now, you can't generally have two parties managing the lifetime. We ask that people who are using Codat kind of trust us to manage the lifetime. And that's another one of those things where it's not as simple always how each of those individual service providers would ask you to manage that token. And it's another headache that I think we look to alleviate.

[00:21:20] J: When you're gathering data and aggregating data across multiple sources with one API to rule them all, right? Like, that's a big promise that you're making to your customers. I'd love to know what's special about your approach to aggregating this data. And then how do you keep that fresh? How do you continuously keep that promise that it's going to be accurate? It's going to work.

[00:21:45] JDS: Yeah. It's a really hard problem. We have a lot of – well, largest engineering resource in head count is a sign to building the integrations. Pulling information from APIs in the weird and wonderful ways that they want us to pull information from them. And then, also, in this kind of this mapping. Like, this mapping to some standard, some aggregated format.

And I think one of the things that we did early on which was maybe different than other organizations that have maybe attempted this or other standards bodies that maybe stand with this is we never thought we were trying to create a global standard for financial information in its totality.

I think it's a very complicated domain. It touches lots of different operational parts of a business and might cover a huge number of use cases. Breaking it up into this idea of data sources, aefining a ring around a set of use cases, and then working out how to best serve those use cases with that data is really the way we thought of tackling the problem, right? That kind of divide and conquer microservices in data standardization or –

[00:22:57] J: Oh, that's interesting. Are you saying you kind of took a bottoms up approach from the use case rather than trying to come up with some super elegant API data approach from the top-down?

[00:23:13] JDS: Yeah, I think that is fair to say. One of the things that maybe when we're often – and we have an internal body that meets us like a center of practice that kind of guides that domain modeling process, and the top line point. And I think this is quite common generally in Codat, is like who is this serving? What's the benefit? What's the use case? Why do they want it? What's the context? Because that's going to lead us to making sure that we build something that standardizes well across a broad set for the problem that people want to solve. The problem that we're trying to solve is, as I talked about, some of those products, whether it's like reconciliation, moving data around. It's helping people make good risk decisions and things like that.

The problem we're trying to solve is not a global standard for financial information, which is a massive problem that supernational organizations have tried to define. And, yeah, I think bottom up for us is the –

[00:24:15] J: That has been my big question through this whole – as I've been looking at your product and reading your documentation. Finally, I think that's the right answer. Or that's a very satisfying answer to say that it's a domain-driven. It's use case driven from the bottom up. Because I always just get worried when the key to success, as everyone agree, is on the data standard.

[00:24:40] JDS: Yeah. I don't know whether you're a web comic fan. But there's a famous three-panel from XKCD, which is like, "Oh, there are 14 global standards. We really need to try and find a way to aggregate these into a common shared standard for everyone." And then **[inaudible 00:24:59]** there are 15 global standards.

[00:25:02] J: Yeah, yeah. We used to say, standards are great. And there's so many to pick from, right? I do really believe in standards, though. I don't want to take anything away from that effort. I do think it's always going to be an 80/20 situation in which standards can cover a lot, but they can't really – they'll never cover everything.

This is the difference between software and data. Software is beautiful because it can be correct. You can get an answer, right? I think data, there's always some part of it that's just sort of a hot mess. Some portion. Not for you guys, obviously. You've got everything completely figured out. But there's always some portion that like is harder to nail down than the rest of it.

And so, I like the approach of – that makes sense to me, anyway, of showing a sort of immediate value, use case by use case, rather than trying to have some master plan. That said, let me just make sure I understand it so that if I'm – what I can do using your service is just use a console to connect up all – I never have – As a user, I can use the API or I can just use the console. Is that right?

[00:26:21] JDS: Yeah. That's another approach I think that Codat's always taken, which is this dial between no code and fully API integrated and API-driven. I think that the great thing about that kind of no code offering, and we use custom branding and things like that, and dashboarding and workflows where needed, is that that no code thing allows you to prove value very quickly, which is very useful in a lot of markets. It also allows you to kind of buy into the API-driven experience at a pace which matches your business's ability to build to that API, which is different for every business. And every business has got a different set of priorities on their roadmap.

You want to be able to prove value as quickly as possible with the products that Codat are offering. And then, for us, to be able to better evidence all of the richness that you're going to get through that API-driven approach.

Go back to that Link product you know we're talking earlier. I think that's a pretty good example. And that we've got a totally managed – kind of you can send someone an email with a link in it that will go through this guided, custom-branded experience in their browser, which is fully hosted by Codat. That's like the ultimate no code experience.

The next level of that is to say, "Okay, you want to keep this experience somewhat native." You can use our embedded connection journey, which is a React component, if you happen to be using React on your front end. And you can embed that flow directly into your application.

Now, everything that those two, either the kind of fully hosted out in a web browser versus this kind of React component embedded, all of that is driven directly through our public API. Any aspect of those systems that are presenting an end user with a list of data sources for them to pick and then sending them off on the integration flow, all of that can be powered by things for our API.

And we think that that kind of this journey from fully managed to API integrated for you to be able to add richness for the experience that you have for the people that you're trying to wedge, whether it's your risk analyst who want data fed into a model or to look at an analyst's tool and assess delivered by Codat is this kind of this dial that you can buy into whatever point that makes sense for you to spend the engineering time to get that benefit –

[00:28:59] J: That's great. Yeah. So, you've got kind of – you've cast a wide net in terms of that end user profile. And it's also componentized, so that you can pick and choose the degree to which you're going to – the degree. That's interesting. You've got this – and assess you're doing automated insights. And some of those may feed into very important risk calculations for the downstream organization.

I also hear you saying that you're looking at the topology of all this integrated data and helping unify it into standard classifications, right? That takes a lot of intelligence. Like, machine learning, artificial intelligence. What has been your sort of technical approach? To me, I feel like your whole business is based on one API to rule them all, data aggregation. But this is a step further into machine learning. Is that right?

[00:30:01] JDS: Yeah, that's right. And again, I think we've always started from the bottom up, right? What is the use case? What is the biggest pain point? And that there are two things there in terms of machine learning. Maybe as one, when we're trying to do something predictive. But there's also the fact that we're a data aggregator allows us to make really interesting statistical analysis, which is the kind of meat and potatoes of data science, is when you've got a clean data set. And one might look at Codat specifically. If you wanted to, from a different perspective, look at Codat specifically. It's like a data cleaning organization, right? That we're creating this kind of standardized data on the basis of specific use cases or business behaviors, which is a great thing to be feeding into a model to learn or to make statistical analysis on, of course. But also, doing something predictive with ML.

And I think the benefit of using something like Codat as well is a lot of smaller lenders or lenders that are just starting up, right? The biggest problem with saying we think we could have a great hybrid model for risk analysis is what are you going to train that hybrid model on day one of your business opening the doors?"

If you could use a no code offering to be able to onboard businesses, then you're going to be able to have an avenue to getting data in order to be able to train your model. And in the meantime, maybe you'd want to be using some tool like Codat, which can make predictions on the basis of this kind of aggregated data set that we have. And you get to leverage then the intelligence that we can gain from much larger sets of data –

[00:31:48] J: Yeah, that's interesting. That makes sense. And then, so, in terms of this data cleaning or making recommendations on how to better operate, is that something that, for the accounting, if you're connected to – I think I read somewhere that if you're connected to QuickBooks or something like that, that over time, you guys can recommend better use of the features in your accounting software or your approach to reconciliation. Is that right? Or did I misremember that?

[00:32:21] JDS: Maybe I might be misinterpreting the question. In terms of reconciliation, I think we understand how QuickBooks and Zero operate. What their operating principles are for doing ERP with those systems? And so, where we're fueling, where we're kind of managing that reconciliation, as we talked about expenses or point of sale information, there's commerce. I think, for commerce product.

Yeah, we've got good knowledge and good relationships with those. We talked about as well as integrations change and as their maybe needs or expectations change in their API, that's something that's going to be frictionless for you using Codat, right? Because we understand both how that software thinks or wants those things to be reconciled. And we have a good understanding of –

[00:33:09] J: Okay. So, maybe it gets a little better out of me. It gets a little bit better out of my use of my underlying account. You help me use my underlying accounting software a little bit more accurately or more fully. Okay. Yeah, that was my awkward way of asking that question. So, thank you. I needed some help on that.

Listen then, I want to shift attention to I think one of the interesting things is you guys have made some particular technology decisions based on some particular theses. Can you share with us some of the big technology decisions since you were employee one, which is a special role? What are some of the big tech decisions people probably don't know?

[00:33:54] JDS: Yeah, big tech decisions that people don't know. I think a lot of it was – And maybe I've mentioned this a few times, this idea of bottom up. I think we labeled it as bottom up. But this relentless kind of pursuit of what the use case. Who's the persona? What is the client problem? Let's focus ourselves around the problem statement and build for that.

And I think we've always done that at every level even when we have technical decisions. And I think this is great practice for any business, right? Is to say we all have a shared roadmap. Whether it's technical or product. And it's all based on an outcome, desired – an expected outcome for a client. And a big part of that as well is what do they expect from the system? What do they need it to do? And that's what we should be addressing.

And part of that is kind of building boring, I think. We talk about a lot – I talk a lot about building with boring technology. You get a certain amount of – let's say you get a certain amount of creativity to spend in a business. And you want to be spending that on tackling the client's problems in the best way that you can, right?

And if you're a new business, it's probably somewhere that you're disrupting, or it's a new product category or something like that that needs a lot of innovation, needs a lot of creativity in that kind of problem space in that business domain. And so, you don't necessarily want to be spending all your creativity in some esoteric technology in order to serve that.

And so, I think that's always been at our core, is like, in technology, it's kind of proving the operational expectation can be met. And then looking at the roadmap ahead in terms of that expectation, the operational expectation. And then building the way that you change technology around that. And that's meant that we've had to be incredibly evolutionary in the way that we design our architecture. And the way that we choose technology, we've had to allow for innovation where it's been necessary to be flexible.

And we do a lot of things. We try and favor composition and choreography as those kind of patterns for achieving that. And the integration space is a good one. A lot of – I think on the face of it you would say, "Here's an OAuth 2 – OAuth 2 kind of authorized API that has a bunch of REST endpoints." Surely, that's the same everywhere. And actually, the reality is there's a lot of things that seem the same and want to be the same in their outcome, but actually in their implementation are very different. Until we've built a technology stack and an SDK for doing those kind of things, which is very composable. That allows us to leverage that kind of broad set of knowledge in order to build new integrations quicker, because we've seen that problem before. But also, to be able to maintain the changing way that we want to operate in any particular integration that then maybe –

[00:37:00] J: Yeah. That's really interesting. Yeah, you're really amassing know-how in this like super hard problem of integration because you have seen the problems time and time again. And you're building in your team this know-how. I think in a lot of businesses, even when you solve a problem, it kind of goes away, right?

[00:37:21] JDS: Yeah. Sure. I mean, I mentioned before, this team, the center of practice, we call it the data model committee – Committees at Codat, which is maybe a tongue-in-cheek name for something that started early doors is maybe part of an idiosyncrasy of Codat.

But the data model committee has been around for a long time in terms of guiding this kind of use case-based approach to standardization. And thinking about we've seen a lot of APIs and how people present data that actually the business behavior might be very similar in all of them. But there are many different ways that that can –

[00:38:05] J: That's intriguing. It's intriguing. It's really embracing at part of the process, right? You're really embracing and going into the storm. Whereas, typically, what you do is you'd want to – if you're doing this manually or on one-off, you do it individually. You probably wouldn't document it. And then you just back out of the room and hope you never have to do whatever this is again. And of course, that inevitably leads to problems for the individual business. But I think it's an asset for you guys to really kind of built up an asset of knowledge around the most painful component.

[00:38:43] JDS: Yeah. And again, talking a bit about company DNA. One of our operating principles is boring problems are almost always undervalued. And a lot of those things I think might seem on the face of it like boring or laborious. But actually, there's a huge amount of value in building that understanding and doing it in a way which is useful. And I think that's the core point. It's got to be useful for something –

[00:39:11] J: Yeah, I want to talk more about what makes a boring technology or a boring problem, because I'm very interested in that. But if you are solving a more humble problem, does that affect your recruiting for software developers who want to work on more of these technical, delight, esoteric things?

[00:39:37] JDS: Yeah, I think if somebody is a software engineer that is driven by a bleeding edge sense of a new paradigm for a way to solve the same class of problem? Yeah, Codat's probably not going to be a great place for that engineer.

But potentially, I mean, there's a lot of different jobs that we do, right? Not everybody is building integrations. But I think it would also be – it'd be disingenuous to say that to the job that we have building those integrations is humble maybe.

I think a big part of it is the scope. It's an incredibly interesting area in terms of the scope. You have this multi-dimensional complexity in terms of the number of use cases, the complexity of the data that we're trying to standardize, which is –

[00:40:30] J: Oh, yeah. I don't really take away any – it's hard. Don't get me wrong. It's hard. I hope that didn't come out wrong. I think there's just a whole discussion going on in the developer engineering world that people go through these incredibly rigorous interview processes for software development positions. And then the real job is something very different, right? Yeah.

[00:41:00] JDS: Yeah. Well, yeah, that's fair enough. I think we've always hired at Codat on the basis of finding excellent individuals who are capable of solving the problems that we need to be solved and then bringing on a whole bunch of skills and experiences that we can leverage to find global maxima of things we might be doing that we could be doing better, that we could be optimizing better. Or there might be some systems approach where actually it's not optimization, it's reorganization that's going to lead us to a better outcome.

And so, yeah, our hiring has always been done in that way. You're not going to sit down and interview at Codat and be asked to solve some obscenely complex graph traversal problem in the least, the smallest, O notation you possibly can. Because that's not generally the sorts of problems we're solving.

And actually, if you've spent a lifetime perfecting that, that's great. There're loads of places where that's hugely useful. But actually, while our world is changing as influx, I think we need to be finding people who have a systems approach. And we talked about that use case. A use case implies some commercial understanding as well. And I think, yeah, those are – in terms of the engineers we're hiring, I think those are the ones that really fly that are looking to – We often talk about kind of heads-up engineers, right? Engineers that are looking up from the keyboard across the business about things that we could be doing better and problems we could be solving or not solving.

[00:42:46] J: That's interesting. I think I'm definitely hearing a theme. And I'm learning about a little bit more about Codat's culture and the theme. Certainly committees. I get that. That's an important phrase. Transparency, this ability to, using your term, focus on boring problems and boring tech, which I want to get back to what you mean by that. And then this idea that it's really very use case domain-driven. I think these are sort of – I'm getting a better picture of the culture at Codat.

Since you were founding – one of the first – you're the first employee. Is that right? The first? Okay. And now how many software developers do you have?

[00:43:33] JDS: Just up to 150 –

[00:43:35] J: Okay interesting. I'm hearing about this idea of having heads-up developers. Being very transparent. I'm just curious, how has that journey been building an organization like that from really just you? And now, that's a pretty big team. Is there any like things you learned or advice you'd pass on to someone building an engineering team?

[00:44:00] JDS: Yeah, I think a few of the things that we've done over time are we quite like growing into an organizational structure. I don't think that necessarily means us creating some what if. Sort of either way of organizing the business or organizing technology or something. But it's about having an understanding of where we're going. Looking ahead and organizing for the next inflection of scale while you can still experiment. I think it has been very helpful.

The other one is early doors and right from the beginning of an individual's tenure, as soon as they start at Codat, is building a sense of ownership, and a sense of autonomy and a sense of systems thinking. I don't think we had an engineer that's – Well, every single engineer that's ever started at Codat has released code production by the end of their first week. That's at least – I think, quite often, it happens much earlier than that. And that's about kind of reducing the fear and the friction to be able to know that they have agency in both making technical decisions and submitting code to production to solve the problems.

And also, that the process that we use, that growing into and having a sense of experimentation, is that every individual is an agent and has agency over changing the process. I like to talk about the fact that Codat is very strict on the rule of law. You must follow the process. We ask that everyone follows the process as best they can. They measure what outcomes are necessary in their business unit. And they follow their process in order to achieve those outcomes. If the the process that they're following doesn't achieve what they want, change the process, right? And be an agent in doing that. And have a bit of systems thinking about where risk can be spread.

And I think those sorts of things helps us keep this idea of startups often higher. You hire a lot of generalists who are good at kind of adapting to situations. And that's the thing that often gets missed, right? Is as a process gets larger, it's going to feel much more rigid. But I think if you build flexibility into it, that the process is there to achieve an outcome. That outcome is the most important thing. Like, the client use case, the client experience. Those things are the most important thing. If you're not doing that faster or more accurately –

[00:46:38] J: Yeah, that's interesting. I was the founder of a startup that got up to 80 people. And I do think you can get a little – in the focus on operational – making operations repeatable, right? Because when you're smaller, it's a little bit more of an adhocracy. You're improving – Improvisational. But then you kind of over-index on process and the way it should be. And so, I think that's right. That mindset that keeps the entrepreneurial mindset of exactly – like, what's the end point here that we're solving for? As well as like, okay, we need some processes. I think that's a really cool – that's an interesting observation and quite accurate.

[00:47:18] JDS: Yeah. And I think sometimes that means that you have to make some sort of trade on efficiency in order for there to be the capacity there to kind of to adapt. And actually, that slight marginal efficiency, it's spent Innovation. And knowing that when people are finding a better way, a better global maxima, that you have a way of disseminating that.

And so, going back to that committees thing, right? We have a lot of working groups and committees, which this whole idea of this bottom-up centers of practice that exist in order to be able to share, understand and disseminate, "Okay, what's your context? What were you trying to solve? What did you measure? And what's the new outcome?" Right? "Should I adopt that or shouldn't I on the basis of those things?"

And we use things like – in tech, we use things like ADRs to document decisions. And we have an RFC process as well. That means if people have a problem that they want to collaborate on, they can publish that to the kind of –

[00:48:28] J: Right. Right. I like that. You keep all the – You add the standard process. The rule of law as you say. But in your opinion, you would – if you had to over-balance towards innovation, you'd probably do that. Yeah, that's interesting. That's interesting.

[00:48:45] JDS: Yeah.

[00:48:46] J: I'm scared of hurting somebody's feelings. But I want to ask, what do you think about a boring technology? Don't name any names. But one –

[00:48:58] JDS: Oh, I'll name some names. I think they'd be proud of it. I think they'll be proud of the moniker. What does boring technology mean? It's like a glib clickbait title. I think for saying that there is a decent level of operational understanding within the individuals who are going to be operating that thing at scale or at the desired scale anyway.

And so, we're a .net shop. There're a lot of engineers writing C# code that's running .NET. And we're on latest and greatest on that six, seven, which is a great run time. That's an exciting run time. But **[inaudible 00:49:40]** Microsoft it's not so boring. But it's been pretty solid. We're happy with it.

But there's a lot of engineers out there who have a real really good understanding about how C# and .NET will behave at scale in a production environment, which means that hiring for those individuals, sharing information, being able to Google a problem –

[00:50:01] J: Oh, I'm so glad you mentioned that. I'm sorry. I'm going to interrupt you one second. Just, say, for investors in particular, you might like – I've often gotten the question, "Well, why do I care what is written then? Or why do I care about these technical decisions?" But it has an incredibly powerful punch line in hiring.

[00:50:23] JDS: Yeah, hugely. And operations, you know? Because you might find it really easy to hire a whole bunch of people who want to use the latest and greatest. But if they're not core contributors to that, not one release of that particular technology, then getting operational understanding of what to do next when you come across something that's either a scale inflection or some particular operational problem. Either what is being seen as a defect.

You need to know where the boundary is when you find some defect in your system. They're always going to happen with software. But you want to know that is it me? Or is it them? Is it the technology? Or is it me? And being able to eradicate that kind of is it the technology question, because that's a much harder one maybe to answer. Like I say, unless you're maybe the key contributor and understand every line of code for something that you're buying into the business. Then you want to be spending again that innovation time, that knowledge and that creativity and finding a way to tackle that defect and learn from it in your business domain.

Yeah, we use C#, .NET. We like Microsoft SQL Server. Again, there isn't a problem you're not going to be able to Google for anything that happens at actually quite a heavy load. It's a system that's been tried and tested at a massive scale.

And so, we can get a lot of leverage of that. And I think there's – like I say, maybe there's some operational inefficiency that you might say, "Maybe for any particular use case or problem, you might be able to find a technology which is more directly suited to it."

But then you've got to look at what is the long-term cost of maintenance for that. That's not only the long-term costs of maintenance in terms of your understanding. That's assuming you're understanding or the global knowledge base doesn't change as well. And how many times as an engineer said they've looked back at code they wrote six months ago and I would re-engineer that completely from scratch.

Now, let's say that that's the first six months you've been using that brand-new technology. There's an extra dimension to that complexity in terms of maintenance and in terms of incurring technical debt. It doesn't mean to say that there aren't places where that operational efficiency doesn't have a crossover point between what's the complexity involved in us building with this technology that's going to give us this marginal difference.

But I think the idea is that you're starting from a point of making sure you're building to the expectation of the customer in a way that you can iterate your software to get there and then starting to think about what's the operational margin that we want to be contracting in.

And we found in most places actually that crossover point at which there's some marginal gain at re-architecting gets further and further away. It's not Moore's Law that's driving it further away given, right? It's not every year the CPU is getting faster. But there's a whole bunch of other things going on in the world around us, right?

The move from HTTP 1.1 to HTTP 2 everywhere. Actually, suddenly, all of the things that are API calls between services, which even at a data layer quite often might be using HTTP or something, suddenly gets **[inaudible 00:53:47]** just because the nature of the protocol underneath. And I think that happens a lot –

[00:53:53] J: All right. Proudly wear the moniker of boring technology. That's what I've learned. No. That's really interesting. Well, I've really enjoyed talking with. I'm just fascinated by your product. It seems amazing. It is offering – I'd love to see small and mid-sized businesses get access to all the same data and benefits of that data that we see in really large companies. The only question I would kind of wrap up with, I'm just curious, are you coding? Or you engaging with any other technologies just to stay technical?

[00:54:34] JDS: Yeah. I guess I'm a technologist. I think a lot of people who have started building from a young age and have continued to can't help themselves. We don't necessarily use it at Codat. But I tinkered on the bits and pieces in Rust. And I –

[00:54:54] J: Okay. I know all I need to know about you now. No. Just –

[00:55:00] JDS: Yeah. I like the functional approach. I think it lends a lot to a different way of thinking about things. In terms of – and I'm always going to be tinkering in that way and building.

I think how that informs the way I think about technology at Codat. I think they're somewhat related.

Do I write code every day? Sometimes. I think there's a good amount to get blamed **[inaudible 00:55:24]** still on being employee number one. I think it wouldn't be decent of me not to lend myself to either correcting wrongs, or sharing contexts, or something like that. Yeah, sometimes I can be found – Yeah.

[00:55:40] J: I was just thinking about your statement there. It's impossible.

[00:55:42] JDS: [inaudible 00:55:42] like you think you could have done something in a better way. And we've been at Codat for five years. There's plenty of code around –

[00:55:49] J: Software engineers and are often musicians. And that's for a reason, right? They're constantly twiddling and going back. That's interesting. Yeah, we should do a show on Rust actually. A lot of stuff – Yeah.

[00:56:07] JDS: Yeah, I mean, that whole space, I think, Go, Rust, Zig. I think a lot of people are trying to solve that the latest C++ versions aren't going far enough for the sake of a nice building experience. A lot of it is to do I think with package management, dependency management. And that seems to be a default now. You need a tool chain, which is cohesive. And I think that's brilliant.

It's nice to see innovation in a space that is – I don't know. What? 60? 60 years on or more? Yeah, it's amazing. Like I said, I started my life in the embedded world writing VHDL, C, Objective C. I still have a heart string connected to C –

[00:57:07] J: I agree. Well, maybe we'll do another show on that. In the meantime, thank you so much for your time and your insights on building. I thought it was just great to catch up with you. And we didn't have a chance to talk – Oh, I'm sorry. Actually, maybe we'll cut this part out. I'm sorry, Jason. I did have one other question. How is Codat doing? I didn't get a chance to talk about the business side. I understand you just expanded into a big new office. And so, it seems like things are going great.

[00:57:36] JDS: Yeah, things are going great. We spent a long time growing out of the office space that we had through the pandemic. And we've just kind of exploded in a great new space. We're still in Farringdon in London, which is where our main office is. We also have just moved our New York team into a wonderful new office.

[00:58:02] J: Oh, I walked past your office in New York. I walked past that. Yeah, just recently, and thought of you. Yeah, yeah, yeah.

Well, well-deserved. I love the boring use case bottoms up approach. I've learned a little something about your culture as well as your technology. And just really appreciate your time, Jason.

[00:58:22] JDS: Yeah, no worries. Thanks for having me on and –

[00:58:25] J: All right. I'm going to stop the recording.

[END]