SDS PODCAST EPISODE 957: HOW AI AGENTS ARE AUTOMATING ENTERPRISE DATA OPERATIONS, WITH ASHWIN RAJEEVA

Jon Krohn:	00:00	What if your data pipelines could fix themselves? Not in some distant future, but right now detecting errors, rewriting code, and redeploying without waking up. an engineer at 3:00 AM Welcome to the SuperDataScience podcast. I'm your host, Jon Krohn. I'm joined today on the show by Ashwin Rajeeva, co-founder and CTO of Excel Data, a startup that's raised over a hundred million dollars in venture capital to bring you the AG agentic data management platform. Ashwyn's outstanding communication on this meaty technical topic involving autonomous scouring over petabytes of enterprise data. Makes for a tremendous episode. Enjoy.
	00:37	This episode of Super Data Science is made possible by Dell, Intel, Fabi and Cisco.
	00:45	Ashwin, welcome to the SuperDataScience podcast. It's great to have you on the show. Where are you calling in from today?
Ashwin Rajeeva:	00:50	I'm in the Bay Area. Usually work out of India, but I'm here four months in a year. And thank you Jon for calling me to the podcast. Happy to be here.
Jon Krohn:	01:00	Yeah, yeah, yeah. So the company you work at Excel Data, you guys are a big deal. Correct me if I'm wrong on this, but I was looking this up before the episode. It looks like you guys have raised over a hundred million dollars in venture capital already.
Ashwin Rajeeva:	01:12	Yes, we have. So we've been around since 2019, founded in 2019 and over the years we have been fortunate enough to work with some of the best investors, raised over a hundred million dollars over three rounds, and also worked with opportunity to work with some of the biggest Fortune 500 kind of customers. So yeah, life's been good.

Jon Krohn:	01:35	Fantastic. Congratulations on all of that early success. And we're going to talk about the Excel data product obviously, but something I wanted to touch on really quickly before we got going that's kind of interesting is it seems like Excel data is more or less headquartered in the Bay Area, but you mentioned how you work from the Bay Area four months of the year and then most of the rest of the year. I guess you're based out of India. If our research is correct, it looked like maybe three out of your four co-founders are based in India most of the time.
Ashwin Rajeeva:	02:04	Yes, all four of us are from a technical background. So we used to work at this company called Hortonworks. Hortonworks may use of course the Hadoop platform along with Cloudera. And once we started, the whole idea was that, hey, let's just put something together quickly start working with customers because we have been working in data for such a long time that we kind of know whom do we need to sell to in some sense. So the three of us, of course started off there in India and we kind of built the whole tech team and platform over there and rohith our CEO essentially moved to the Bay area and somewhere in 2020 to set up the GTM and sales and all of it. And most of our engineering comes from that background of big data specialists and engineers, data engineers. And so we found that it's the best place to set up an engineer org. But we have in the Bay area as well. We have some engineers here, we have some engineers in Canada to be close to the customers in the East coast. And we are setting up a presence in the EM EEA region as well because we have a small team in London that we are setting it up.
Jon Krohn:	03:15	Very nice. It sounds like you guys have things figured out all over the world. Let's talk a bit more about what you're up to there and the products that you're developing. So as a CTO and co-founder of Excel Data, you've been driving the shift toward a agentic data management,

which I think you guys call a DM for short, and it seems like a key part of that is something that you call the X Lake reasoning engine. Do you want to tell us about X Lake and ag agentic data management in general?

Ashwin Rajeeva: 03:44 Yeah, absolutely. So I think before we get into what Ag Agentic data management is and what this whole X Lake theory is, I think it's a little useful to talk about where we're coming from. We work with the bigger banks, insurance company, the telco companies, and usually what's ended up happening is that this field of data management and components in it, so if you have a data catalog or MDM curation, any of it, any of this stack, the practices which have been around have been now around for 15 years, most of the companies have been around for that period of time. I'm not talking about the compute engines because they come and go, but let's say the field of metadata and MET management is essentially now the same for the last 15, 20 years with incremental improvements. And what we thought is as AI becomes more and more capable, we've found that I think a lot of work which is manual and businessy because in the end a lot of business software is a lot of forms and a lot of clicking and a lot of working with interfaces, with very well-defined workflows.

05:01 And we feel that with the advent of ai, a lot of that can essentially be automated. And so there are two parts to it. One is the A DM part of it, which is the how do you make data management agent in the sense, let's say you are a data steward and you are doing five things regularly in your day job. How do you essentially make agent steward? That's where the industry's heading, whether it's sales or marketing, everybody's trying to do this. So we are trying to apply that to data management. So can a data lake essentially classify the data itself? For example, can you create validation rules automatically without having to spend an inordinate amount of time on some of

these things? And what we also realized is that to power something like this, you actually need an engine. So a simple example is that when you use an ai, it knows everything about the world because in its brain there is a compression of facts and almost the whole internet, but it knows very little about your data.

06:07 So if you ask it a simple question, a DM, a simple question saying, okay, look at my data and let's say create a report about some business topic that you have, it really has no way for it to go and figure out what's in your environment. It only knows about what's the internal logic it has. And so what we've done is created a crosslake compute engine, which essentially provides the tools to the AI layer, which it can use to connect to your internal enterprise systems. And we call it an X Lake reasoning engine because the way our architecture is built is that you can actually connect all your data sources or data lakes together. So whether you're on cloud, you might have big companies have all sorts of environments. They will buy Snowflake, they will buy AWS, they also have Azure, they also have a huge on-prem footprint. So why should we call it the X Lake reasoning engine is because it can provide AI the context it needs while operating across different data lakes. And that is the reason why a DM is so powerful that you can almost ask it anything about your environment and it knows how to get data from your cloud, from your internal environment, from your data centers, put it all together at scale and provide the ai, the real tooling or the context it needs to answer your question. So that's what that's all about.

Jon Krohn: 07:39 Nice. All of that makes perfect sense to me. That was a great explanation of the problems that people are facing and it makes so much sense to me that in data management enterprises are trying to do the same kinds of things as they're doing in marketing and sales and

software development and be able to automate as much as they can. This is the future of business, there's no question about it. In our research we pulled up that, I don't know how accurate this is, but we pulled up that there are four particular enterprise data agents that you guys have that solve existing data problems and unlock new capabilities. Does that make sense? I'm talking about these kinds of four particular types of agents.

Ashwin Rajeeva: 08:17 This is very interesting. Every company or everybody who you talk to solve talks about agents and essentially something with agency which can act by itself. That's the whole idea. Now, what we've also realized is that there are aspects of any enterprise where it's easy for them to adopt new technology like agents. For example, let's say your salespeople have a call and there's an agent recording that call creating summaries and sending it to your AEs or something like that, which makes a of sense and it's easy to do, but the same thing cannot be applied to maybe some sort of discount offers being run on a data lake and then reports being created automatically and then shown because the business problem itself is fundamentally kind of customized to each company. So if you're working with a bank, they have different problems. If you're working with insurers, they have a different problems.

09:15 The healthcare companies has a different business problem. So it's not easy for us to create some sort of a stock data management agent and say, okay, it's applicable to everybody. So what we're doing is talking to our customer piece and essentially trying to figure out what makes sense in the data management world and distill it down to maybe 4, 5, 6 agents. Right now we have kind of centralized, I think some of the material online is old, but we are now almost close to 10 agents. And these are to do with different aspects of data quality. So I'll give you an example, right? One of the most important things

that a data enterprise does is essentially create a catalog of all your data. So we've created a data catalog agent, which essentially does the same, but it has the knowledge and the context which is derived from all your metadata.

10:09 So it knows if you ask for a sales order table, where is it, how many columns it has, what is the data type? It knows almost everything about it. When does the data come in? How often does it come in? Who's the owner of it? So it knows everything. And the same way we have created, we are starting to create all these different agents. And so when you ask a question to a DM, it can kind of spin off these agents and go and figure out what can be done and how can I best answer what a user wants to do.

Jon Krohn: 10:39 So it's sort of similar to maybe when I am in the chat GBT interface or I'm in the cloud interface or Gemini and I ask a deep research request and it goes off, it spins up some number of agents. It seems like some of these platforms like chatt T seems to spin up just a few agents. Anthropic seems to spin up like hundreds of research agents and then they go over the public web. Or if you have connected things like your Google Drive or Microsoft Office or whatever, it'll look over all of those kinds of things as well. So it kind of sounds like a similar kind of idea, but this is operating in a way that is enterprise grade and it's designed for looking over, as you described, X Lake all of the Exactly.

Ashwin Rajeeva: 11:23 Exactly. And so that's a very interesting thought because I think in your example, you just nailed it saying, Hey, you ask a question to Claude and it says that, Hey, I don't have this information in my matrix, so I'm going to go and look at the internet and look for the top 10 results and try to break you something right now, whichever search engine it uses, it's done. Let's say it uses DuckDuckGo, right? Or it uses Google. Now they have done the hard

work of integrating all of the Internet's information into a searchable index and given anthropic an API, which you can hit to say, okay, I'm going to look for, let's say, which talks in my portfolio need to be liquidated because of whatever reason. So it's going to do all this research, hit the Bing API or the Google API, and get information back. So Google has all done all the hard work to provide you with that data, but not just imagine an enterprise. They have petabytes of data across many, many different data lakes. And so even if you connect a cloud, so the first question is how do you connect a cloud interface or a charge repeat interface with an API, which can actually aggregate data over this petabyte data scale? And how do you do it at scale?

12:44 For example, you ask for some dataset saying, Hey, aggregate around my sales order, how many different segments I have, and now in the end, this has to be translated to a cluster level query, which has to be fired in your data center in some parquet format. And so that's what the XLE interface does. It essentially provides you the tooling required to access your data lake and provides it in a way that AI can make sense of it. So we do all the hard work to provide you the index to your data store.

Jon Krohn: 13:21 It's pretty wild. As you described that process to me, I'm trying to wrap my head around it. When you're talking about petabytes of information unstructured across all these different data lakes and being able to bring a query back to your users quickly, I mean, that's pretty wild. Congrats ashman on getting that all together. Yeah,

Ashwin Rajeeva: 13:39 Yeah, it's hard. I mean, it's a problem which has to be solved. I think it's going to be solved not just by us, but over the years because this is this often repeated thing that the AI is as good as the data, it has access to thing, but I think it's a little confusing because the model, no matter which model you use, whether you use an open

source one or you use the state of that foundation model, it can only access the information which is compressed in its neural net. And that information provides it a lot of effective ways of doing things. For example, you can ask it to write an SQL query, it'll write it, and these days it'll write it absolutely correctly, but once it's written it, it needs to hit somewhere. And when it needs to hit it, it needs to hit it with the same the scale. It has no idea how much data it's going to return. It has no idea whether the user who's using the interface can access a column in that query. And so that's what we want to do. We want to make sure that the X Lake engine is like the context provider to AI models over your entire data landscape. So that's the whole idea behind it.

Jon Krohn: 15:00 That's really cool. Thank you for digging into that specific SQL example. And it reminds me of something else that came up in our research, which blew my mind and requires so much nuance to get right, is that we read about self-healing data pipelines through these kinds of age agentic workflows that you have. So this is autonomous pipeline optimization that allows self-healing without human orchestration. So how does that kind of system work? Obviously without spilling too much proprietary sauce for our audience, but how does that work in a way that prevents hallucinations or divergent interpretations from cascading through the system?

Ashwin Rajeeva: 15:38 Yeah, it's a great question. I think easiest describe with an example, let's take a data pipeline. Most data pipelines are one of two ways. You have some sort of a drag and drop interface, which you drag and drop an ETL, and then you create some sort of data pipeline, but usually it's a part of a larger structure and maybe a return in a form of in DBT or some sort of airflow. Let's pick Spark for example. Spark Technology writes a data pipeline. Now Spark in the end is code, right? And so let's assume that you have some piece of code, a return in spark,

which is representative of your data pipeline, pick data from somewhere aggregated, make some joins, and dump it to a different place. Now, what will end up happening is that this spark code is being executed often onto your cluster right?

16:34 Now, let's take one step back and let's look at how is something like agent coding working these days? So we can pick up, let's say visual studio code or Cline or Cursor or something like that, and we can ask it to generate code. Let's say we create a small website or a stock follower, anything. And so it's going to generate a lot of code and it's going to use your laptop or your Mac OS as the runtime because it's going to say NPM Run, right? It's going to run something and it's going to use your laptop and its CPU and its runtime to essentially show the browser. And then you say, okay, I like it. The design is great, everything is okay, now I'm going to deploy it. And then you create a deployment out of it and you put it on a server where that website uses the resources or the runtime of that server, not just imagine a data pipeline.

17:29 The data pipelines in bigger companies are not executed on laptops and servers. These are usually executed on clusters of machines. So whether it's a Hadoop cluster or a Trino cluster or a Kubernetes cluster, and then you're running jobs and you're executing your data pipelines on it. Now what we can do is let's say you get an alert from your data pipeline which says that, hey, there's some issue and it requires a small fix. Now, before ai, somebody had to file a bug report and you had to clone this code, and the person who knows about what this code does and what needs to happen can actually make the fix, put it into a CI, and then go deploy it on the cluster. That's how it's going to work. Now, in this day and age, what you can also do is that you can have an agent detect if there is a log entry which shows some sort of an anomaly or something is not right, you can actually clone the code,

automatically, provide the context of this error, provide the metadata of what this data, this pipeline is touching through the X Lake engine saying which table, how many columns, what data type, and the AI can actually then just rewrite this spark code, and then you can then go ahead and deploy it back into the execution engine, which is the cluster, right?

18:54 Theoretically, all of this is possible, and because we know that AI can generate great code, especially with Gemini three, almost one shot, most 90% of what you want to do. And then if you automate the rest of it, which is the metadata context, how do you deploy it into the cluster? Then you can imagine an agent which can actually self-heal, right? And that's somewhat the path which we have chosen is how do you provide number one AI context about where's the code, what is the pipeline, what is the metadata, and what are the errors which are happening? And then attempt if the AI can actually fix the code or the pipeline by itself. Now, not everything can be fixed if you have some proprietary Informatica pipeline, HVR pipeline, some Oracle standard stored procedure doing it not so easy. But more and more, especially with now with AI because of code generation I feel is one of the biggest applications which we'll see, it is probably number one in terms of the potential because the code can express a lot more than anything else

Jon Krohn: 20:04 For sure, and you can know if it works or not.

Ashwin Rajeeva: 20:06 Exactly.

20:08 And so more and more we feel that data pipelines are going to shift from drag and drop interfaces to code because now it's as easy, it's maybe easier to generate code than actually drag and draw pipelines. And so as more and more data pipelines become code, then it's easier to also mutate that code through AI when you have

an error. So all you need is something which gives you the context of your entire data lake, what are the errors happening or are the issues happening? And then essentially have a pipeline which kind of feeds it. So that's the part to autom remediation. Now, I won't claim that we can auto automate everything, but if most of the factors are right, hey, you have a code-based pipeline and it's largely it's version controlled and it's spark or it's SQL or something, then it is entirely possible to attempt some of these things.

Jon Krohn: 20:57 That's really exciting. And everything that you were talking about there was fully autonomous, but it sounds like your age agentic data management system, A DM also allows for a human in the loop if people want that. So how does that work? Do you define when a human should be in the loop or does the client define that? Is it on a case by case basis? How does that work logistically?

Ashwin Rajeeva: 21:21 Yeah, philosophically, I think even the best agent systems or even the simpler ones, you would want some control before you actually accept it. Whether it's you probably won't trust an AI to generate automated emails for you, you would probably want to review it. And I think the more complex the problem is, I think it's going to be very hard for people to accept anything blindly, which an AI provides. And so when we talk about autonomous, I think the work differential which you can generate versus a human being, that's the autonomous part. So it's as simple as let's say you ask a colleague to write you up some code or a module and you do a code review before actually accepting it for most every junior engineer or anybody in your org. And it's a similar philosophy. I think the differential, the autonomy, the value, the economic value comes from the fact that you've compressed maybe one week of work into maybe an hour by doing a lot of autonomous stuff.

	22:33	But I think the need to approve disapprove, review or reject has to come back to somebody in the value chain. And usually these are the people who are the experts. So somebody who knows the system but now does not need, let's say an army of engineers, but he needs an army of agents or they need an army of agent, but they still have the control on reject review, do better or just completely ask you to rewrite. I think that's not going to go away. And the way I think about it, that's what autonomy means, not just that you do the whole thing without any supervision, but you do most of the work without supervision, but there's somebody at the gate all the time.
Jon Krohn:	23:17	Of course. That makes a lot of sense. And so this complex system, this autonomous data management platform, a DM that runs across petabytes, petabytes of data, has agents involved, has these human in the loop capabilities, it sounds like for the most part, if not entirely based on our research, you can correct me where I'm wrong here, but it sounds like a lot of this you guys did from scratch. You didn't want to rely on existing open source technologies and just kind of build on top of that.
Ashwin Rajeeva:	23:48	I think some of it came from the fact that we've been around since we was 19, like I said. And so we've built talking to customers, we've built a lot of other technology as well, which can then go ahead access data, different engines. How do you reliably run, let's say validate a million rows every hour for quality, something like that. And we have essentially ended up using the same technology under the hood, and especially in this field, I think what differentiates us is that we are not just a metadata player. A lot of, I think data management relies a lot on metadata and then providing value on top. Our engine differentiates itself is by also having access to the data underneath. So when you deploy the ADOC or the A DM platform, you just don't have something on the cloud and then it sucks all your metadata.

24:45 You also deploy what we call a data plane, which is in your environment. And we have built the technology to provide a secure MTLS system between the two. So no matter how much data you ask for, you can ask us to validate a hundred rows, a million rows or a hundred million rows, we rely on the data plane, which runs in your environment to actually have access to data. And so over the years we've had to build a lot of these pieces. How do you connect them, how do you make sure the jobs are running reliably? How do you authenticate users down into somebody's environment? And so in some sense, we have kind of put all of that together and reused a lot of that stack. But of course we wouldn't be there without open source technology. So we built out our stuff on Kubernetes Park and all the good work the community does. And we've also tried our best to give back some of it. So we run an open source platform called ODP. You can actually download it. It's like a full data platform, which is available for you if you're interested. And that is something which we have built and self-manage because a lot of work, our work internal work is also making sure we can run different environments at scale.

Jon Krohn: 26:01 That's really cool. ODP, it's just like a GitHub repo.

Ashwin Rajeeva: 26:03 It's just a GitHub repo. Yeah, it's our own version of the whole big data management stack and it's open source and we patch it and we build it. We put the security vulnerabilities there. A lot of these kind of primitives get used at the underlying system. If you have, let's say a rack which is available to you and you don't have compute, you can use this. This is open source compute up to date with upstream vulnerability is taken care of, and it all integrates with the accelerator platform. So it's more of an end-to-end story that we have built. And when you start up, let's say you're four people, you're 10 people, you don't have the resources capital, so you pick your battles. So we have done the hard work over the

years to release different parts of the component. And now we feel with a DM, we kind of have the whole stack. So we have a DM, we have aoc, we have the data management, we have the compute as well at the bottom, and now we have ai, which can actually accelerate a lot of this work as well.

Jon Krohn: 27:08 I see. Found the link to your open data platform, ODP. So I'll have that link for our listeners in the show notes and our viewers in the show notes to check out. Thanks for that. So let's go back a little bit to the way that you have built this platform, A DP to span petabyte scale, complex, hybrid multi-cloud environments. In a previous interview, you've described this as something like the captain's deck where you have all the information you need across this extremely complicated system at your fingertips. And you noted that part of the big problem that you're solving with Excel data is what you call data sprawl. Do you want to tell us about what data sprawl is? That's actually not a term that I think I've ever come across.

Ashwin Rajeeva: 27:54 Yeah, it's very simple. I think I once did a round table and a lot of these representatives from most enterprise companies come in, and I keep repeating this term enterprise is because I think it's very important to know the kind of customers and the people that we work with. Technology is only as useful if it's applicable. So it's very hard for us to sell our software to another startup because the startup is busy building the business. They're not in a space where they're investing hugely in data technology. That's the game, which requires money, requires time, and you need to be in a position where you are using the data for something. So for example, a customer of ours, like a Nestle is a global organization where they invest huge amount of money in data to provide competitive advantage, right? Different markets and all the skews that sell.

28:53 It's not true for let's say a company like us, which is a hundred million. So some of these problems are enterprise problems. And one of the things which an enterprise usually does is that they buy a lot of technology. So if you ever have the chance to talk to some of the data leaders, they will always tell you that they have been doing this since the time 2010, build data platforms. And so that environment is never homogenous, right? You won't find a company which is essentially frozen on one technology. And there are different groups in the organization and they use different set of technologies. So now usually you end up having some cloud environments where there's some data, there's probably some investment on something like a Snowflake or Databricks to drive analytics. Maybe some new leader came in and invested in new technology. Of course, there's a huge legacy landscape of data sitting in data centers.

29:52 And that's what essentially data sprawl is. You have hundreds of terabytes of data on your on-prem clusters and you moved some of it for analytics into the cloud, and you moved some of it into Google because you signed the cloud transformation journey with Google. And now you end up in a situation where you have data everywhere and somebody needs to find out what's where, how do I access it, and maybe put them together in some sort of a report. And that requires you to, number one, know where all the data is. And number two is to be able to access it in a good way, which can make sense to a user. And so that's what really data sprawl is, right? It's the nature of, it's the, let's say, eventual fate of any big enterprise that they have data everywhere and they want to centralize, but it's hard. It takes many years. Until then they need to work with all of this data.

Jon Krohn: 31:00 Fantastic. Well, this has been a great tour of the Excel data platform, all the exciting things that you have going

on there. The a agentic data management platform, A DM as well as the ODP. Yeah.

Ashwin Rajeeva: 31:14 ODP.

Jon Krohn: 31:14 Yeah, open data platform as well. And so I'd like to shift gears here a little bit now to maybe some more generic insights that you might have for our listeners. Of course, it'll still end up centering a lot on Excel data, I'm sure given how much it's been a big part of your life for the last six years. In an interview you mentioned realizing that being a CTO is as much of a people problem as it is a pure technology problem

Ashwin Rajeeva: 31:41 And

Jon Krohn: 31:42 That you had to move from an architect slash programming mindset to putting yourself in the shoes of customers, of marketers and employees. So what kinds of guidance do you have for our listeners? What kinds of habits or frameworks should our listeners be thinking of to have the kind of success that you've had as a technology leader and growing a huge startup like Excel data?

Ashwin Rajeeva: 32:07 Yeah, we've grown from four people, 200 people. I'm just talking about engineering as such. We have of course more people in the on and we have done this through COVID and remote work and then coming back to office. The reason why I said it at that time, and I really believe it is because most of the time as technologists and programmers, you always believe that your version of what you're building is correct. It's the best way of doing something. So if I build it, of course they'll come. That's the standard thing. And then there are two aspects to it. One is that why do you need to think of it from your customer's perspective? Because the customers, they operate in a different sort of an environment. They don't

have the same degrees of freedom that you do. And most of the time everybody is coming in from some sort of a guidance.

33:06 So the way I think about it's that a new CEO comes in or a new data leader comes in and they come in with some sort of vision that, hey, I can do this because I know I've been successful somewhere and now I'm going to go down this technological path. And then somebody down the organization gets a guidance and the guidance is it has two access. One is the technology access of it, which is saying, okay, we need to modernize, we need to use this technology. And for example, now the whole thing is we need to do ai. That's the whole thing, something like that. And the other one is the business, which is that there is a business problem you're trying to solve. This is not like a school project where you build some sort of a website and say, Hey, this Gemini one shot at this and it's so cool.

33:52 So usually it's in those axis, and unless you actually put yourself in the customer's shoes and see if your technology or whatever you're building is actually applicable to what they're doing, can bring them some value, can make them successful so that they can show this success to their leadership and then it actually adds value. So next year when they come back to you, they're willing to buy more or they're willing to continue with your P-O-C-P-O-V and then convert it into a contract, and then you get revenue and then you expand from there, you build new offerings. So unless you think about it, because your engineers are not going to think about it. So you hire people, you give them a task, you say, Hey, you're going to work on ai, you're going to work on data, you're going to work on network. Nobody's going to think about it.

34:45 So somebody in an engineering org has to actually do it. I've always felt that product managers are great, but

they're focused on features and issues and bugs, and it's really difficult to be that good at all of these aspects. So as a technologist, somebody in the A, so if not the CTO, then who else has to think about the fact that am I building something which will bring value to a customer? And how do you create an org which kind of gets that information? And so that's from a customer's standpoint and from an engineer's perspective, most people would not be, this is something which I've read as well, right after some point a salary or money is not so interesting to people. And so if you want to attract people who want to stay with you, especially during a startup's first few years, you don't want people to come and go all the time because it kind of hinders your ability to carry on context and build at a certain speed. And so if you want to bring in people at an early stage, they have to have the belief of, Hey, what is this company going to do? Why am I doing it?

36:07 What's my way of expressing, let's say creativity through code or through engineering? And that is hard to do if all they get is a bunch of Jira which have been done through sprints. So you have to create an environment where people think that they are in some sense innovating in whatever domain the company has chosen for itself. And for us, that has to do with enterprise and data management. And so the way we want to run our r and d and engineering is to see how much we can innovate here and not just the fact that hey, the company roadmap has been decided in the beginning of the year and here is a bunch of plans for the rest of the year and do it. So we have allowed or let's say work with our team to innovate on a lot of things. And some of these examples are like the ODP link, this is completely engineer led initiative.

37:02 If you don't have a data platform, we needed one internally for testing. We don't want to pay license visa, we don't want to have security volume. So we built it. We

also built our own vulnerability management systems because we ship a lot of this enterprise software and everybody will scan it. There's a lot of vulnerabilities. So we built a whole AI agent to fix vulnerabilities and we've built our own kind of data center stack like an OpenStack because VMware is expensive, OpenStack is too complex. So we built something. And so this is I think what inspires people to say, Hey, I'm here to do technology and this company happens to do technology in this arena and that's where I'll spend time. So you got to put yourself in the shoes of the engineer you're hiring and also the customer you're talking to and then try to bring in a plan which kind of works for both of them. So at least that's the way I think about business.

Jon Krohn: 37:56 That is such a great answer. You went into a lot of detail there on even the kinds of things I was going to ask about in my next question, which were, because you've said in past interviews that a talented programmer is looking for meaning in their day-to-day that highly paid engineers still just complain about their jobs. It's funny to me to think that somebody who gets a hundred million dollars signed bonus to said meta is then just like, oh man. But I'm sure that happens. I don't know where the stat is at today, but when I was doing my PhD, something like 15 years ago, I attended a lecture on the economics of happiness just for fun. I just went to this lecture and at that time it was showing that in the us, if a household was making over something like 80,000 or a hundred thousand US dollars a year, that is the happiest you can be making more money beyond that point. Did it make people happier? Now with inflation, the numbers are probably a bit little bit higher, but directionally, I think this kind of gives the idea that you're explaining, which is that beyond having your basic needs taken care of and knowing that you have security for you and your loved ones, the extra money beyond that could end up being a hassle. Yeah,

Ashwin Rajeeva: 39:12 It is. It is. And it's also interesting. I mean there are studies on developer productivity and you would see that the numbers are insane. I mean, people talk about how an engineer, a software engineer is productive, may be four hours a day or three hours a day, and the rest of the time is spent in meeting planning, whatever. It's right now I feel that even if you take two hours for meeting, you still are leaving a lot of this time out. And if you think about it, what better privilege can someone have than to sit in usually a great office on a laptop without having to move moving is optional and then get paid top dollar for it. And most engineers then leave their jobs. It's not just people leave ator, but people leave all sorts of companies. And there has to be a reason for it is that most people would be happy because knowledge work is something which has to do with creativity.

40:12 It's hard to sit in one place and realize that the work which was presented to you or asked of you could be done maybe in two hours. And then you've got to sit and find something to do and it's good for a few days and you spend some time, but after a while you start getting this feeling that, Hey, what am I doing? I'm supposed to do something better. Let me find a mission which resonates with me and my work and my philosophy of it. And so I think making sure that no matter what the business environment you are in as an executive, the engineers or the r and d teams believe that fundamentally we are in the business of innovation in this field. And there are very few fields, whether it's something data management, even something as boring as they enterprise content management. I'm sure that there is innovation that could be done new ways of doing things and people should believe that they have the freedom to do it and they're not just dictated by quarterly plans.

41:15 And this, I think if we can provide an environment like that, then new ideas come in. And for us it's worked out

because for a company of our age and size, we have a lot of, let's say, capability that we have built over the years, whether it's do with ODP, aoc, we have a pulse monitoring system, we have a DM, we are working on the next version of our platform, which will be released in May. And so that is what allows us to do it is where people believe that hey, in this field of what the company has chosen data management, there is innovation that can be driven through pure engineering work. And that's what drives people.

Jon Krohn: 41:58 Nice. I like that. How do you say in an interview, do you think you have a way of telling whether somebody's going to be passionate about a technical infrastructure heavy mission like data management at Excel data versus somebody who's just coming to collect a paycheck?

Ashwin Rajeeva: 42:14 I think it is easy to tell in some sense. Of course, we have made mistakes there as well, like everybody else. But I feel once you start talking to people, and this is what I felt, I mean I've always felt that management in the technical field can only be done by people who have been in the trenches to some sense. And I'm sure there are models everywhere else and which are different. And people have seen managers who work extremely well without actually being on the field. And so the number one thing, at least when I look for potential hires is to see if they can build things. And it doesn't have to be working on some data problem. The question is can you build if you are given a problem, and it could be any problem, it could be something like, Hey, how do you design, let's say a e-commerce warehouse? How do you design a logistics system or any other business problem? And then can you translate it to something which you have learned?

43:39 You probably know, go Python or Java. Can you put something together which presents a real world problem? I think if you can, then those are the people who bring

the most value, who can actually look at a business problem and then convert it down to what they know. And that's what technologies are good at. Of course, this is for slightly senior people. I think for people who are just coming out of college, it's purely based on potential saying, Hey, some of it is your scores and your background and some of it, Hey, how interested are you into doing this? And then you take a bet and maybe after a few months you decide, but for most senior people, I would recommend checking if they can build things.

Jon Krohn: 44:20 That makes a lot of sense. When you're interviewing today, to what extent do you encourage people or discourage people from using LLMs to support themselves? And how do you assess if you are in a situation where you are trying to assess somebody's capability in Go or Python? Yeah. How do you do that and ensure that they're not using an LLM to cheat?

Ashwin Rajeeva: 44:42 Yeah, so one of the things which we've done now this year onwards, is not to have remote coding interviews. We don't want to do it. There are way too many tools which you can use to cheat and it's very hard to tell. And I always felt personally very uncomfortable trying to when I'm talking to you, but I'm always looking for an evidence that you're doing something wrong. And so it takes away from that conversation, isn't it? And so we don't want to do that. So the number one thing which we're doing now is face-to-face. The second thing is that we tell the candidates that, Hey, of course you're going to use LLM CT job. We have given everybody cursor licenses and charge GP licenses, whatever you need. But this interview is all about problem solving. And so there are two things we have done. One is to go towards more practical approach where we kind of created a problem, which represents what you would do day to day.

45:39 So a bunch of failing tests, some missing implementation, and you won't do it. Of course you can Google, right? Nobody memorizes all the APIs. So you can Google figure out, but don't use an LLM to answer the question. And even if you end up using an LLM, unless you install cloud code or something, it's not possible to cheat because you'll have to just copy paste context from different places, find an answer. So make some sort of a gentleman's agreement with them and then start off. But I feel that more and more code will be generated by LLMs, so might not be a scalable strategy going forward, but this is what we're doing right now.

Jon Krohn: 46:19 Yeah, it is really cool how much you could be getting done with LLMs Today organization. And on that note, a few months ago you demonstrated an MCP server that prompted A-G-D-P-R compliance scoring agent and an interactive dashboard, and it highlighted how AI could complete in minutes, what would've historically taken a team days of coordination, research estimation, and meetings. So as a CTO, building a long-term product roadmap, how do you reconcile this compressed innovation cycle that's now possible today with the slower risk averse realities of the large enterprises that are your

Ashwin Rajeeva: 46:58 Clients? The enterprises, I feel they're all slow in themselves in the decision cycle, but they want it because nobody wants to lose out on innovation. So that's why everybody's invested, even though they probably won't be using it or they don't have the go ahead from all their compliance and security people or whatever, but they've started using it internally. And so I feel this trend more and more is going to continue. And at some point, some of these people are going to adopt it faster. They will see a competitive advantage. The other people will be like, okay, we are left behind. Let's go figure out what we need to do to get there. And over a period of time, it will become more and more easier. And so our job is also to anticipate

the future in some sense. We say we don't build something because somebody wants it today, but you look at where the industry is going and you realize that AI is coming for these sort of jobs, let's say as well, or these sort of functions more appropriately and then plan to meet them there.

48:06 So that's the whole strategy behind it. So the whole MCP demo, which I did was kind of a idea which says that there's this engine which has access to all your data, then you have a coding environment. Now can you connect the two and create something which would've taken somebody a long of time to do, but an AI can actually do it now it's not complete, but it's not incomplete as well. So the AI has done say 60% of the work and you could probably prompt it to get 80% of the work, and then maybe two engineers can finish the rest instead of having a whole team. So more and more, whether it's vendors and technology builders like us, or it is people who are adopting AI in enterprises, they will realize that unless they do this, they're going to be left behind. And we want to be in a situation where we meet our customers when they are ready, and that means we have to do all the hard work of making sure we are ready as well with the solution.

Jon Krohn: 49:04 Great answer. Ashwin, as have all of your answers in today's episode, I've really enjoyed listening to you speak. You're a wise man, if you don't mind me saying it's really nice, really enjoyable and outstanding communication. So really enjoyed this episode. Ashwin, before I let my guests go, I always ask for a book recommendation. Do you have anything for us?

Ashwin Rajeeva: 49:24 Yes, I do. So I don't have a regular reading habit, but I try to do at least a book in a couple of months or so. And I was super interested in understanding about, especially the G two as they're calling it, the whole US China rivalry

in some sense, which is going on. So I read a couple of books, but I think the most recent one, which I'm almost done with is called House of Huawei, which is by a lady called Eva or Eva Lu, which I forget the name. And it's all about how Huawei started and how it became the technology company. It is, and also goes in the beginning of China's transformation from a kind of socialist, completely socialist to now a capitalist plus socialist society, and also addresses the challenges around some of the human rights stuff. Very interesting. Highly recommended for anybody to read just to get what's going on.

Jon Krohn: 50:38 Yeah, that's really interesting. I hadn't heard of that book, but it sounds like a good mix of technology, geopolitics, news, just being able to understand what's going on in the world better.

Ashwin Rajeeva: 50:47 Yes, it's a great book.

Jon Krohn: 50:48 Nice recommendation. Thank you. Alright, so for folks who want to get more of your insights after this episode, where should people follow you? Do you have social media accounts that people should follow?

Ashwin Rajeeva: 50:58 I'm a little bit active on LinkedIn about especially accelerator and what we're doing. I don't unfortunately have an account on X or anything like that, but LinkedIn probably is the best place. And then Excel data has some channels, so please visit our website. And there's our resources and blogs and stuff we have built and written about, which you can read.

Jon Krohn: 51:18 Yeah, we'll have links to all of that in the show notes. And I got to say, I don't think you need to apologize for not having an X account, people saying today on this podcast that they just have a LinkedIn account is the most common answer. That's how things have evolved in recent

years. Yeah, a few years ago it was different, but now it's crazy. It's LinkedIn almost all the time.

Ashwin Rajeeva: 51:40 Yeah, it's LinkedIn. Also, I feel people are a little measured in communication over LinkedIn than they're on X. And that has to do with how the algorithm and what they want the platform to be, which I think is more of a tabloid than a source of information and somehow it doesn't appeal to me.

Jon Krohn: 52:05 Yeah, yeah, for sure. I think it's interesting. LinkedIn, it's almost because it doesn't really change very much. I think people get comfortable with that. And it's funny, social media platforms, they always feel like they need to be changing and adding this and adding that. But I think people have found that just being able to see often business related information that's relevant, but often still also entertaining as well. And I think a little bit positive. So I think something that's interesting, you talk about the algorithm there over at X, it seems like it wouldn't surprise me if on LinkedIn I am regularly interacting with people who have very different political views from me, but it doesn't matter at all. You don't even notice.

Ashwin Rajeeva: 52:51 Yeah, yeah, exactly. And X is different. Yeah, I think it's a personal preference. Some people are that way that they are, they would like to engage in a debate and maybe it's not just all about and they find joy doing it. And for people it's more of a broadcast where they put out their thoughts and then maybe some people find something interesting in it. But I think one should be able to, and it's something which I'm thinking about as well, that you should be present as a representative of the company because you don't know where customers are, what people find interesting. So nothing to do with X itself, I think it's just that some people like me just not

		comfortable with that sort of engagement. That's it. But we should be doing more.
Jon Krohn:	53:45	Yeah. Yeah. Alright, well thanks for that unexpected little social media conversation at the end. I really enjoyed today's episode. And yeah, wishing Excel data all the best, seems like you guys are on the right track. And yeah, hope to be catching up with Excel data again soon on the show.
Ashwin Rajeeva:	54:03	Thanks John. Thanks for having me. I had a great time and hope to be back soon.
Jon Krohn:	54:10	That was an awesome episode. I learned a ton. I hope you did too. In the episode, Ashwin Rajiva covered how Excel data's agentic data management platform uses AI agents to automate data quality checks, cataloging and pipeline maintenance across enterprise environments. Compressing work that traditionally took weeks into hours while keeping humans in the approval loop if desired. He talked about how their X Lake reasoning engine solves the problem of AI models having no context about your internal data by providing tools that connect to data lakes across on-premise cloud and hybrid environments, enabling queries at petabyte scale. He talked about how self-healing pipelines work by having agents detect errors and logs, access the code repository, understand metadata context through the X like engine, rewrite the spark or SQL code and redeployed it automatically to compute clusters. That's pretty crazy to me. And he talked about how engineering retention comes down to meaning and innovation rather than compensation alone.
	55:06	With studies showing engineers are productive only a few hours per day when reduced to executing predefined sprint tasks instead of solving creative problems. As always, you can get all the show notes including the

transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Ashwin, Reva's social media profiles, as well as my superdatascience.com/957. And thanks of course to everyone on the SuperDataScience podcast team, podcast manager Sonja Brajovic, media editor, Mario Pombo, partnerships manager, Natalie Ziajski, researcher Serg Masís, writer Dr. Zara Karschay, and our founder Kirill Eremenko.Thanks to everyone on the SuperDataScience podcast team for producing another super episode for us today for enabling that super team to create this free podcast for you. We're so grateful to our sponsors. They allow this show to happen. I guess you guys do too by listening.

55:58 We're all working together here, but you can support this show by checking out our sponsors links, which are in the show notes. And if you yourself are interested in sponsoring an episode, you can find out how at john cron.com/podcast. Otherwise, share, review, subscribe, all that good stuff. But most importantly, just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, keep on rocking it out there and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.