1:19

Alexey

**This week, we'll talk about becoming a data engineering manager. We have a special guest today, Rahul. Rahul has over 12 years of experience in data and engineering. He has been a manager for the last two, almost three years. Now he works as a data engineering manager at Siemens. Welcome, Rahul.**

1:39

Rahul

Thanks Alexey. It’s a pleasure to be here.

1:43

Alexey

**Yeah, thanks for coming and for joining us. Before we go into our main topic of becoming a data engineering manager, let's start with your background. Can you tell us about your career journey so far?**

# Rahul’s background

1:56

Rahul

Yeah. So first of all, thanks, everyone, for joining. And thanks, Alexey, for inviting me. Yeah, I started my career somewhere in 2010 – 12 years back and started as a traditional ETL developer. I was working with databases (relational databases) like SQL Server, Oracle, ETL tools like SSIS, SSRS, Micro stack, and then I slowly moved into more master data management kind of stuff for enterprise. Somewhere in 2015, I started getting into cloud. The volume of data was increasing, so I got some skill set to deal with Big Data and that got me into more of a data engineering role.

I made the transition from a traditional ETL developer – or a BI developer, as you can call it – to more of a data engineering role. Nowadays, I’m dealing with a lot of batch data and real-time streaming data, building pipelines, exposing data to consumers – to downstream systems. It's been a very interesting journey in the last 12 years and currently. In the last two and a half to three years, I took up this data engineering manager role where I am managing a team of enthusiastic data engineers here in India and building an IoT data platform for one of the Siemens businesses.

3:32

Alexey

**That's a pretty cool journey. Your journey into what we call today “data engineering” started with cloud adoption, right? You saw that many companies started to use cloud and this is when you turned to the transition from ETL developer to a data engineer. Do you think it's different or just the same thing, but under a different name?**

3:56

Rahul

It's absolutely the same. I mean, the basics remain the same, but you have to acquire a new skill set when this data paradigm is shifting. A lot of things change when the volume and variety of data grows. In the traditional days, we were dealing more likely with structured data, which is more relational in nature, but these days with evolution of machines, sensors, and social media, we are dealing with a lot of unstructured and semi-structured data use cases.

4:29

Alexey

**Before you probably used tools like Informatica, Talend, Microsoft Integration Service, and now it's Kafka, Spark, and things like this, right?**

4:41

Rahul

Yeah, it's more custom solutions these days, which fits our needs. We are also moving toward more open source frameworks.

# What do data engineering managers do and why do we need them?

4:52

Alexey

**Okay. Yeah, thanks. So let's talk about data engineering managers. What do they do and why do we need them?**

5:01

Rahul

Actually the core still remains the same. As a people manager, you have to take care of your team. You have to keep looking at how your team is moving, your deliverables, et cetera. But another thing that adds up to this role is stakeholder management. When you're working as an individual contributor, you just care about your piece of work. Here, you have to talk to various stakeholders who are sponsoring this data platform and who are also consumers. These days, the number of and variety of consumers are also growing. They are the stakeholders for us.

As a data engineering manager, you have to take care of them and their needs. Again, your team is also a stakeholder, so you have to make sure their skill sets are getting upgraded along with the onset of new tools and technologies that come into play and then coaching them – deciding their career path and sometimes other things. For example, deciding for some approach within a team, or sometimes conflict resolutions. One of the most important tasks for a data engineering manager or any engineering manager should be prioritizing the task. You have a lot of things coming your way from various stakeholders. How you prioritize that is one of the key responsibilities as well.

6:32

Alexey

**So you work on an internal data platform and you manage the engineers, right? The setup we have at OLX is – we have engineering managers and people who report to the engineering managers are backend engineers, frontend engineers, and data engineers. We don't have a role called “data engineering manager”.**

**Usually, there's just an engineering manager who manages the engineers. Now I'm wondering why we have it this way – maybe because we work in so-called “feature teams” which is a team that works on a specific part of a product and everyone works on the same thing. While in your case, your team takes care of the data platform. So everyone on your team is a data engineer, right?**

7:24

Rahul

Absolutely. Correct.

7:27

Alexey

**Yeah. I guess it's helpful for data engineers to have a manager who is also a data engineer, or at least someone who can relate to the problems they have. This way, they can think about the best way to approach their development and their coaching. Whereas if we have just a regular engineering manager, they might not know about new data tools in detail, and so on. That's where you come into the picture – you give suggestions to people and tell them what is good to learn, right?**

8:03

Rahul

I think you mentioned a very valid point. This role, when I talk to other engineering managers, I realized that as a data engineering manager, you have to be in line with some hands-on activities as well. You should know what your team is working on at the code level sometimes. That's how I spend my time. I wear two hats here.

50% of the time, I am working as an individual contributor – as a data engineer in my team – and the remaining 50%, I'm managing the team. As you mentioned correctly, sometimes when you have technical brainstorming or maybe defining roadmaps, then your technical skill sets really come into play. A data engineering manager should definitely have some hands-on knowledge, and not only just with managing people.

8:54

Alexey

**How do you find time to actually do this? How large is your team – how many people do you manage?**

8:59

Rahul

Currently, I manage eight people. Including me, it’s 9 people. [laughs]

# Balancing engineering and management

9:05

Alexey

**[laughs] How do you find time? I guess if you just have one-on-ones with everyone on the team, half a week is already gone. So how do you manage to find time to actually work on hands-on activities?**

9:16

Rahul

Yeah. Actually, it says it has been changing. Initially, when we started, the team size was small. I was able to do justice to both of the roles. Now, since the team size is growing, sometimes it happens that this ratio [between the roles] becomes 60-40. But I am fortunate enough to have a team around me that is self-motivated and self-organized. As part of Siemens culture, we promote servant leadership – not micromanaging our team, but rather enabling them to do their best. This means that when they need to, they can come to me or come to any of my seniors – my managers.

Also, when we hire, we look for someone who can own the piece of work with minimal support from a manager or senior folks. That way, the team is more self-driven, I would say. This is how I can find some time for myself to work as an individual contributor. But I know – I agree with your point, once the team size reaches a certain threshold, it becomes difficult to perform both roles together.

10:33

Alexey

**Basically, if I understood you correctly, you have some quite experienced people in your team, who can take care of helping others with technical problems. And in general, the team is quite self-organized, meaning that most of the time, they don’t need you. Then you can focus on other things. That's pretty nice to be able to have such a team that you can be sure of and that can just work independently. If you need to work on something, you trust them to find a solution to whatever problem they have. That's pretty cool.**

# Rahul’s transition into data engineering management

11:09

Alexey

**How did you become a data engineering manager? How did it happen for you?**

11:14

Rahul

Interesting question. Earlier, as I mentioned, when the team size was smaller – we had 4 data engineers, including me – at that time, we all used to report to my manager, who is currently my manager. Since the business started growing and the need for the data platform grew, we proved some of the use cases that the data platform can solve. We were getting more and more use cases. Naturally, the team size increased as a result. At that time, my manager, who is my current manager, just called me to a one-on-one discussion and asked if I wanted to take up the engineering manager role. My reaction was “Why me? We have other folks.”

Basically, what I realized is – he mentioned a couple of qualities which he saw in me. I was frankly not aware of those things. He said I have the ability to see the bigger picture – what business problem we are solving, along with some business acumen and domain knowledge, and situational understanding “What is needed and when?” That kind of quality he saw in me, and that’s why he asked me this. He also mentioned, “Once you pick up this role, you have to sacrifice a few things.” We are all passionate about writing code and building stuff, and deploying it, but if I take up this role, then I have to sacrifice at least 50% of the time that goes on these things. So that was my natural evolution – I was hired as a senior data engineer, but immediately after six months or so, I took up this engineer manager role.

12:56

Alexey

**Interesting. So the qualities you mentioned are being able to see the bigger picture, knowing what the business problems are, and you also mentioned situational understanding “Knowing what is needed and when”. Do you think that for somebody who is a senior and who wants to become a manager is it enough to have these qualities?**

13:15

Rahul

Well, I think these are the essential ingredients, but one thing that I forgot to mention is about people skills. That is most important. In one word, I would say “empathy”. That is something that is more of a behavioral factor. So whether you can empathize with the team or people who report to you – that's one of the most important ingredients. Then you add these things that I mentioned as well as some technical skills and business knowledge.

13:54

Alexey

**What do you think about technical skills? Are they important for this role? Or are they less important than empathy, seeing the bigger picture, and knowing the business needs and other things?**

14:07

Rahul

I think they're equally important for this role, because this data landscape is changing very fast. We are getting more and more new problem use cases every day. I have seen in past teams that managers are not able to cope with the recent technology and trends, which are needed to uplift their data platform or data products.

In data engineering management especially, you have to always uplift yourself in terms of technology stack. You may not be going into too much detail of solving every individual particular issue, but at least the high-level idea – the limitations and features of particular products or offerings – you should be aware of.

# The importance of updating your skill set

14:54

Alexey

**Okay, so let's say for an example – right now you heard that there is a tool called Prefect, which might be nicer and then maybe you allocate some time to give it a try. Then you see how it is different from the current technologies that you use. Then maybe you also show others, “Hey, this is the tool I played with. Let's give it a try.”Or “Let's not give it a try, because it doesn't really work for us.” So you’re talking about doing things like this, right? Just trying new tools and then sharing them with the team.**

15:25

Rahul

Yeah, absolutely. Along with some senior data engineers and the team, you just brainstorm what problems this tool can solve. It has various angles, like the cost involved, licensing fees, how easy it to work with, etc. – so all sorts of analysis. As a data engineering manager, you should be able to contribute to those kinds of discussions. That's why I think having technical know-how is not negotiable.

15:59

Alexey

**For you, I think it's especially important because you spend 40 to 50% of your time on hands-on activities, right? You wouldn't be able to do this if you didn't have these technical skills.**

16:11

Rahul

Absolutely. That's why consciously, I chose this. I have the flexibility here in Siemens to become a full-time people manager without giving much time to the technical aspect. But consciously I chose this way – to be in line with technical trends.

# Planning the transition to manager and other challenges

16:32

Alexey

**So you had this chat with your manager and he told you, “Hey, you're doing such a good job as a data engineer, how about becoming a manager?” Your first reaction was “Why me?” But then I guess it took some time for you to think about this and agree. After that, what did you do next? Did you do anything to make this transition smooth, or what did you do?**

16:58

Rahul

Not exactly, to be very honest. I didn't jot out any plan, because the team was growing at that point in time. Though I did get involved in some hiring, etc, but I did not have any plan at that point in time. It was more of a natural evolution. But over the last two and a half to three years, I have learned many things that I was not doing initially. And I have also learned a few things that I could have done better when I took up this role. At this point, I would say it was more organic. I learned on the job, you know?

A lot of the time, I used to do one-on-ones with my manager in order to understand how I should be thinking – not as an individual contributor, but as an engineering manager. But when I reflect back, I think I could have done better planning, like the 30-60-90 days plan or something like that. Then it would have been a much smoother journey. But still, it was quite a decent journey for this transition.

18:14

Alexey

**So learning by trial and error, right?**

18:18

Rahul

Yeah, sometimes. A few things I have learned the hard way, which I could have avoided. But it's fine.

18:25

Alexey

**What were the main challenges for you in this transition?**

18:33

Rahul

At the beginning, the first challenge was to get accepted in the team. Because you are one of the team members who just elevated as a data engineering manager. So building trust with the existing team and getting accepted was the biggest challenge initially. Then later, I realized I was getting into all the stuff, each and every piece of work without prioritizing them. It was really overwhelming. That’s another thing I learned – how to prioritize. When you're dealing with multiple stakeholders, people are coming from everywhere. There’s too much context switching that you are doing. It's difficult to focus on certain things. This was the biggest challenge initially.

Over a period of time I learned, reflected back and learned again. You cannot achieve everything on your plate, so you have to prioritize the high priority items, critical items, and then go with that. One more challenge, initially – I spend a lot of thinking about it – how do you uplift your team? That's one of your core responsibilities. Currently, if my team is delivering or staying at a particular quality of individuals – next year, am I able to uplift them or not? Are they getting better or not? Those were the initial challenges.

20:11

Alexey

**How about delegating? Was it difficult for you to do? You said that you realized that it’s impossible to achieve everything. You cannot do everything yourself. How did you realize, “Okay. I cannot do this thing. I need to delegate it to somebody else.” How was this for you? Was it difficult? How do you learn to do this?**

20:33

Rahul

It was difficult and it still is difficult at this point as well, to be honest. The culture we have here in Siemens – you do not dictate things to people. It's more of an open culture. In this case, influencing somebody to take on a few things is really tricky, but I'm still learning that and I have learned a lot in recent times. This comes from trust – if somebody has trust in you and they can just take on a few things which are maybe not interesting in nature and they do not like doing it, but just to support you, they will do it. So, I’m still learning, but that was a challenge initially.

21:19

Alexey

**You said, if you were to go through this process again, you would do better planning – you mentioned this 30-60-90 day thing. Can you maybe tell us a bit more about that? What else would you do differently if you needed to go through this transition again?**

21:41

Rahul

Yeah. Again, when I reflect back I see that I was initially playing it very safe. I was not questioning people if something was not working well and just trying to build some trust. At the time, I was not questioning people and not being very assertive in the beginning, because I didn't want to counter them with certain questions. I thought the relationship would be good if I did it this way to start with. But I think there are a few things where I could have been more assertive with the team and with some of the stakeholders on some non-negotiable items.

That’s something that I would do if I got the chance to relive those moments again. Another thing which I think I should have done in the very beginning is setting out clear expectations with the team – defining core goals and objectives etc. Also how together as a team we will achieve those goals or how they will achieve them as individuals. I started that a little bit late, but I think I should have done that much earlier.

# Setting expectations for the team and measuring success

22:55

Alexey

**And how do you set clear expectations for a team? Let's say you have a team that works on a data platform for IoT. How do you set expectations for them? How do you understand what the team's purposes are?**

23:15

Rahul

Yeah, it took quite some time to build the framework to set this because the nature of businesses changes very dynamically and you will have the requirements that change very often. What I did and we did as an organization and as a team is define the goals by placing them into two categories. One category includes committed objectives or goals, and the other category is aspirational goals. In the committed objectives, we put certain things that should not be missing delivery timelines – no spillover into the next sprint, the code quality should be good, no defects – all those things that are expected of software engineers or data engineers. Those are committed goals that are non-negotiable – ones that you have to achieve. We have certain metrics to track those.

You also have to uplift the team. With a mature team, all committed goals are more or less met always, but the challenge comes when you want to uplift the game. Goals like “How do you modernize your data platform?” Or “How do you challenge the status quo of the processes that you are following?” Those are the goals that I put in as part of aspirational targets or aspirational goals. These aspirational goals should be big enough that you should not be able to achieve 100%. That's why we call them aspirational goals. If you achieve 50-70% of that, then you're good to go. That way, we are trying to build a framework for expectation management and performance management.

25:04

Alexey

**Thanks. We have a related question from Lok. The question is “What are the success measures of a data engineering team?”**

25:13

Rahul

As a team, I would say success is measured by how you enable or cultivate the data culture within the organization or among the stakeholders who are interacting with you. Before data came into picture, people made decisions based on the judgment calls or gut feelings – those should be more data driven in nature. That, I think, is not a tangible thing, but rather it's a cultural thing.

Something tangible is if you are able to serve more and more use cases for our data consumers and by removing some of the frictions, meaning how easily we can expose our data to various stakeholders in different formats. It could be like reporting, visualization, some RESTful API's for applications, some data for machine learning or data science community within the organization. The variety of use cases you are able to support with the use of a data platform, I think that's one of the KPIs of success.

26:39

Alexey

**Interesting. Are there any other formal metrics? We have objectives and then you probably look at the objectives to understand if the team is moving in the right direction. So apart from that, maybe there is no need to have other formal metrics and KPIs. Right?**

27:03

Rahul

We capture some metrics. The data quality metrics is something we capture. We also have the data reconciliation framework, which we capture in terms of percentage – for different lines of businesses of data. Along with that, some basic percentage coverage of our source system data – what we have to offer – and how we are enriching the data. This means how we uplifted the quality of the data using some smart ways, including third party data and things like that. It's very difficult to measure the success of a data platform because it sits at the back end. There are certain limits to what you can put as the metrics. But I think one of the key metrics is how many consumers you're serving and if they're interested in your data.

# Data reconciliation

28:04

Alexey

**Yeah, that's a good one. What is data reconciliation?**

28:09

Rahul

Basically, you're pulling data from various source systems and loading it into your data warehouse or data lake. In between that, there is an ETL layer involved. What we do once our data load is done in batch mode or real-time mode, we reconcile what we had in the source system versus the target to see “Are we losing any data in between or not?” That’s the kind of framework we have built.

28:34

Alexey

**Yeah. You’re not supposed to lose any data, right?**

28:39

Rahul

Absolutely.

28:40

Alexey

**What are some reasons for why we can lose something? Because of downtime, right?**

28:46

Rahul

Because of downtime or because of leakages in your ETL pipeline. You're filtering some data – let's say you're not putting proper exception handling sometimes – that's how you can also lose data.

# GDPR compliance

29:01

Alexey

**There is a question that is related, I think, to the data platform. “Data engineering is moving towards full automation with data versioning and managed model serving – how do you solve GDPR compliance issues and bring stakeholders on board?” That's quite not an easy question, right? What do you do about this GDPR in your case for IoT?**

29:28

Rahul

Absolutely. We do not have many GDPR-related compliances issues. Our business is B2B – it's not really to an end customer. But still, we cater to certain data sets which have to be GDPR-compliant. Basically, the data warehouse tool that we have chosen is cloud-based, where we do a lot of dynamic data masking, which is a lot different from the static data masking in traditional days. We have role-based access control put on top of it, so only certain people can see certain kinds of data. We have tagged our data in terms of “highly classified” and “public” data sets, and then put dynamic data masking and role-based access control on top of it. Therefore authorized people can access the data which they're supposed to.

30:32

Alexey

**“Dynamic data masking” is when you see some personal data or some restricted data, then it is replaced with something else. Right? So instead of an email address, you see a hash or something like that?**

30:45

Rahul

Yeah, hash, cross, asterisk sign, etc.

# Data modeling for Big Data

30:50

Alexey

**Yeah, nice. Thanks. Another question from Nishikant is, “How do you handle data modeling for big data? Do you do Agile or something like this to handle evolving requirements?”**

31:06

Rahul

Initially, we had quite a lot of challenges with this because we have defined a certain data model and the variety of use cases started increasing. Earlier, we were following typical ETL methodology, where we extract the data, transform it, and load it into a target data model. But that was very tightly coupled. The data model was very fixed in nature. That's how we realized that we cannot live with that. It's not scalable or flexible.

We changed our approach from ETL to ELT – basically, pulling the data, loading into our data warehouse, and then transforming it on the go. That's how our data model became more resilient, more flexible in nature, and with less dependency on different data marts or tables. We have also implemented a data lake for our data platform, which is more robust in nature compared to a data warehouse.

32:11

Alexey

**So, if something changes, then it means that maybe you will need to go back to the previous year and recompute some things if you do a change in schema, right? Because you keep the raw data in your data lake, so if you make some changes in your data model, then you just go and recompute it.**

32:29

Rahul

Yes. We do not change data models very often. Rather, we have a flat data model in place. We serve the consumers and perform all these aggregations in real time for most of the use cases. However, certain pieces of data, which are a kind of “master data” are more structured in nature. There, we have very traditional data models. Basically, in order to handle big data we are dealing with the combination of a traditional data model, a flat data model, as well as the data lake.

33:06

Alexey

**What if the raw data changes? I guess you will need to somehow think how exactly you can read the old raw data and the new raw data, right?**

33:15

Rahul

Yeah. We are maintaining data lineage in this case. A lot of time, we put in cache the old data that is most frequently used in our warehouse. When the new data comes, we recalculate that – I mean, invalidate and reactivate the cache to serve the data to customers faster.

# Advice for people transitioning into data engineering management

33:39

Alexey

**Okay. Back to our topic of data engineering management – there are still a few things I wanted to ask you. What would you suggest to somebody who is transitioning right now into an engineering management role and specifically to a data engineering management role?**

34:02

Rahul

I think there are a few things that you have to accept from day one. You cannot continue playing the individual contributor role, because you have limited time. Some of the mistakes I have seen people make more often are that they still are very much fascinated with the coding part of it. But you have to just uplift yourself and see the bigger picture from the whole business point of view, “What problem are you solving?” The second thing is empathy – if you're genuinely interested in the growth of your team members, that shows. It is clearly visible.

You should have certain career planning strategies for every team member. You need to sit down with them, know their aspirations, build some career planning strategies, and enable them to achieve those skills or the goals that they’re aiming for. Just to summarize, I would say let go of a few technical things and don’t go much deeper into all the technical aspects. Then, empathize with the team members. You also have to be on top of the game. You have to know what the latest trends are, so you can enable your team to think in a certain direction or be able to expose these new areas of work to them.

# Staying on top of trends and enabling team members

35:38

Alexey

**That was actually the question that I wanted to ask you – how exactly do you enable team members? One thing is to be on top of the latest trends. How do you actually do this? How do you stay on top of trends? I'm a data scientist and in data science, there are so many things happening, it's just simply impossible to stay on top of it all. I think data engineering is also an actively developing area – new tools appear and new ways of doing stuff appear. How do you manage to stay on top of that?**

36:08

Rahul

When I say “stay on top of it,” I don’t necessarily mean that you always change your platforms, tools, and techniques. You need a certain kind of stability. When I say “reinventing yourself,” I mean getting better at what you’re doing. For example, let's say somebody joined my team as a new data engineer, and it took two or three weeks to develop a data pipeline – to test it, and deploy it into production. Next time, if the same pipeline comes along or a similar thing comes along, it should take a little less than that. That's how you grow.

The second thing is – we have come a certain way in the last two to three years with a majority of data platforms. What are things that we can make more modular? What are some things that we can automate? We have to constantly think about those lines. We have to make sure that team members are not doing monotonous work. That should be the benchmark. When you challenge that sort of monotonousness, obviously, you will try to figure out new ways or smart ways of doing the same work. That's how your team will always be indirectly challenging the status quo.

37:25

Alexey

**Okay, so it's not about staying up to date with state-of-the-art things and knowing all the newest technologies, it's more about knowing what's happening in the team and how we can make their lives easier and the work more interesting, right?**

37:43

Rahul

Yeah. It's sometimes a challenge to put some restrictions on this because people are very passionate about “I want to implement this new tool in the market or automation. Build versus buy.” All those things. But you have to just sit down, calm the team, and just focus on the end goal – what we are trying to achieve. Don’t get fascinated with the technology, new tool sets, or SaaS offerings each day or continuously.

38:12

Alexey

**Yeah, it's very easy to get caught up in it because you see all these tools appear, people on Twitter talk about this and in other social media. It can feel like “Okay, this is the tool. I need to stop doing everything and try this.” But this tool is not actually solving your business problem.**

# The qualities of a good data engineering team

38:36

Alexey

**In your opinion, what are the qualities and skills that people in the data team should know? You manage a team as a manager – what qualities should the team have?**

38:54

Rahul

First of all, I think they should start with understanding the problem statement or the business problem. Sometimes there is no problem, but using the power of data you can solve certain use cases. Firstly, the team should be mature or think from that direction and then comes the technology part. If I talk about the tech stack, I always go to very basics – as a data engineer, you should know SQL, ETL concept, data warehousing concept, at least one scripting language like Python or Spark, a typical CI/CD framework, and cloud. These are the basics that are important – which are non-negotiable in the current situation. Ownership is something that I think is the most important piece.

40:02

Alexey

**What is ownership?**

40:05

Rahul

Ownership is basically a certain decision you're making. If something goes wrong, or maybe something fails – you own that and you do not go into certain blame games. You just try to fix it first and then see what happened without getting asked from managers or senior people or stakeholders.

40:35

Alexey

**So, if something breaks, you don't wait until your manager comes and says, “Hey, why is this pipeline not working?” Is that right?**

40:45

Rahul

Absolutely. Initially, I used to get a lot of alerts when something failed. But then I just delegated that work. We decided on the ownership of certain pieces, so I do not receive alerts anymore. I trust the team to take care of these things.

# The qualities of a good data engineer candidate (interview advice)

41:00

Alexey

**Okay. Technology-wise, you said that the non-negotiable minimum in SQL, knowing what ETL is, knowing what a data warehouse is, one scripting language like Python, knowing the concept of CI/CD, and knowing a cloud platform. That's the technology part. Then there’s also this business acumen part – knowing the problems and things like that. These are the qualities that people in the data engineering team – the data engineers – should have, right? We also mentioned ownership. As a manager, you probably devote a lot of time to hiring people. I guess these are the things that you look for in candidates – you want to see these qualities in the candidates, right?**

41:51

Rahul

Absolutely. Plus there’s one more thing that I would add. When it comes to hiring, especially with this remote work environment, something called communication skills becomes very important. How you communicate and how proactive you are. One thing I look for and the very first question that I sometimes ask to the candidate is, “Explain in five minutes one of your best projects – one that you implemented or that you were part of.” I would say that shows how a candidate articulates things and if they stay to a very precise answer. That is something that was important before this pandemic, but nowadays it has become even more important.

42:44

Alexey

**“Explain your project in five minutes” I guess the “five minutes” part is also quite important, right? Because I imagine if somebody asks me about a project I'm proud of, then they’ll probably need to stop me at some point, because I can just go on and on and on. So here, you have to be concise, right?**

43:05

Rahul

You have to be concise, and just think as if you're explaining these things to non-data or non-IT people. In other words – how would you explain this without going into too much technical detail? First, start with a business use case and what you or your team did to solve it. That's why I keep a limit of five minutes. If I have any further questions – if I'm interested to know something more – then I dig deeper into it and a set of follow-up questions takes place.

43:42

Alexey

**So when you start talking about the project, you should start with the business problem. Do not jump into, “I was using Spark to move one terabyte of data in Parquet from S3 to Google cloud storage.” Nobody cares, right? [laughs] Start with the business problem.**

43:59

Rahul

Absolutely. That's a common mistake that people make. They start jumping directly into the technology part of it. I mean, that's something they are passionate about, but you still have to be able to articulate these things in less time.

44:23

Alexey

**Yeah. This is a mistake that I also make quite often because I'm so proud of what I built and I want to talk about it. But then I forget about the fact that I was not doing it for the sake of doing it, but because there was some problem that I was trying to solve. It's important to bring this up. What else do you ask in an interview?**

44:48

Rahul

Initially, I did part of the technical screening as well, but these days I focus more on the hiring manager round. Somebody from my team or the PR team does the technical discussions. If that goes through, then the hiring manager round comes, where I mostly focus on, as I mentioned, about the project. The second thing is just trying to gauge what the candidate can do – not focusing on what they have done in the past, but what they can do in the future for the team or the organization. It is difficult, but we ask certain incidents and how they would react to them.

Sometimes I put them into certain hypothetical situations and see how they would respond to that. For example, “If you want to launch Amazon.com in Antarctica,” or “Launching Uber in China.” That gives me the thought process of the person that is not related directly to data engineering concepts, but how a person thinks about a problem. The answer may not be 100% right, obviously, but I just try to gauge the approach. Along with that, there are certain leadership qualities, like customer obsession. I ask them to tell me some cases that showcase their customer obsession and their ownership, as I mentioned. So, a discussion along these lines.

46:30

Alexey

**So, what's the right answer to “How do you open Amazon.com in Antarctica?” [laughs]**

46:36

Rahul

[laughs] I don't know the right answer. If Amazon hires me and asks me to do it, I may not be the right person for it. But I think this starts with… There could be many right answers to it.

46:54

Alexey

**Maybe the right answer is that you don't open it there?**

47:03

Rahul

No, no. That's a prerequisite that you have to do it. [laughs] You cannot bypass that.

47:13

Alexey

**Okay. Can you tell us about your best hire?**

47:18

Rahul

Yeah. I think there were a couple of them in recent times. They have shown similar qualities when they went through the interview. As I mentioned, these are the technical skill set plus the hiring manager behavioral skills, which I try to judge people on. When you talk about the “best hire” something has to be different than the other hires, right? I recall one hire about six months back, and another one maybe eight months back. Both of those guys adhered to all these qualities that we were trying to judge them on. But on top of that, they were very keen on understanding our use cases, what we are trying to achieve, and the job fitment, basically. This means “What do we expect from them in the short term and the mid-term as an engineering manager.” So they were trying to gauge that.

They weren’t just applying for a random job and just appearing for an interview, but they did a lot of due diligence about the company and about the team. They reached out to other team members over LinkedIn, saw their profiles, where they come from – all this due diligence they did. Once the interview round was finished, we were about to go to the HR discussion. But, each of them separately, asked to have one more discussion with me, where they kind of interviewed me about their career path, what's the next role they should aspire for in the organization. We also talked a lot about culture. So sometimes that shows the genuineness of whether you want to work for the organization or not. So I would say the due diligence and assertiveness is something that sets apart “best hires” from other hires.

49:20

Alexey

**What I heard from all this is “If you want to impress a hiring manager, ask the recruiters to schedule one more interview.” [laughs]**

49:28

Rahul

Oh, my God. You just exposed this. [laughs]

# The difference between having knowledge and stuffing a CV with buzzwords

49:35

Alexey

**There is a quite related question from Fredrik. The question is about hiring data engineers. I also saw this in my work. We often get candidates who just put a lot of buzzwords in their CVs, like “big data,” “security,” and “scalable.” Then you add a bunch of technologies like Spark, Flink, Pulsar, Kafka. You just put a lot of buzzwords there. Of course, I mentioned that these people pass the first recruiter filter because all the keywords match, but then at the end, maybe apart from the buzzwords, there is no real knowledge. So how would you solve this before the interview as a data engineering manager? Is there a way to somehow spot these candidates?**

50:30

Rahul

Before the interview, it's a little difficult. Because in the screening, you just have a profile or resume in front of you. I think nobody even reads the whole resume, because we don't have that much time. You just glance at a few things and that's it. But regarding the HR discussions, what we did recently is passed on certain questions to HR to just get the idea of it. But if some candidate goes through it somehow, most of the time I capture this during the hiring manager round. These guys sometimes tend to clear the technical rounds as well. But we had internal discussions with technical interview panels regarding how to spot these things.

Instead of directly jumping into raw technical questions, we start with another way. If we find certain things that are not adding up, like certain technology or framework that they claim they worked on, then we ask them to give us a scenario of why they used this technology, or in which scenario they used this kind of technology. Then you can just simply classify if this guy has just mentioned this for the sake of mentioning it or has actually worked with it. This involves asking very basic questions. For example, somebody claims “I worked on a data lake and data warehouses.” You just ask “What is the difference between a data lake and a data warehouse?” Or “on-premise and crowd” these are very basic questions. If they are able to articulate or mention this, then fine. Otherwise, you can spot these things.

52:16

Alexey

**So basically, you ask about the context in which this technology was used. If somebody says that they used Kafka then you ask why Kafka was used for the problem they were solving and what were the alternatives – also why they decided to use this particular technology versus alternatives.**

52:34

Rahul

Absolutely, yeah. The context is very important.

52:38

Alexey

**I remember I was also doing CV screening and I saw that a person mentioned all these Bash tools like, awk, sed, Perl, and all that. On one hand, I really wanted to ask that person, like, “Hey, do you really know all that?” Then, on the other hand, because I don't know this myself and people who know these things really impress me.**

**I wasn't sure if they just put this for the sake of putting this. Let's say, if you don't know a technology, but you see somebody put this in their CV and you know that this is difficult – it's a complex technology – how do you assess if they actually know what they're talking about?**

53:23

Rahul

Yeah, that's difficult. It happens sometimes. It's not very often, but it happens. What I do is, during the interview, I just open up that particular technology or tool that they mentioned – just search on Google, which opens that company website or Wikipedia. That gives what this tool is all about in like two lines. It comes with experience, also. If you do it again and again, you build the habit of understanding a new technology. Maybe not understanding the technology, but understanding the concept of the technology and why that exists. At the same time, I just open a browser and while the candidate is explaining, I get certain questions around those technologies.

54:18

Alexey

**Well, maybe they can also do that, right? Just open the Wikipedia page.**

54:23

Rahul

Yeah, but you can just read a few things from there – the question would not be directly from that Wikipedia page on the website.

# Advice for students and fresh graduates

54:34

Alexey

**Yeah. That makes sense. There is a question from Krushal, who I think is a second-year student. The question is, “What would you suggest acing in order to become a data engineer?”**

54:53

Rahul

For recent college graduates or students, I would say to focus on the basics. Do not jump into any specific tools or technology. Sometimes it happens. It happened with me as well, when I was starting my IT journey. That time, I think Informatica and all these other tools were very famous in the market. So, don't focus on particular tools and technology. Instead, in your college days – focus on basics like, how a database management system works, the basics of SQL, and maybe some Python. Just try to understand how software engineering works at that point of time. Get to know basic terms of what a data warehouse is, what ETL is, before getting too attached to any enterprise tools. Build the basics first and then jump into what the industry demands are later.

55:57

Alexey

**I guess if we talk about a four year program and you're currently in the second year of the program, by the time you graduate, the tools that are popular right now may not be popular anymore, right?**

56:08

Rahul

Absolutely.

56:09

Alexey

**Informatica was very popular, but now – how many people use it? I think they still have a large piece of the market share. But still, I don't think it's as widespread as it used to be. So maybe it's better to invest your time in learning something that is more commonly used. But how do you know what would be on the list in two years? You don't. Right?**

56:34

Rahul

That's the shift I'm seeing these days. The difference between the “great” candidate versus the average candidate you hire. This is a person who is able to reinvent themselves again and again. They just get better again, and again – that is something that sets you apart from other average candidates who are more focused on understanding a technology or tools. If your basics are right and these are three-four technologies like querying, Python, etc, then you should be good in terms of learning any new tool in the market. These days, these tools are very user-friendly, or rather developer-friendly. They're putting a lot of good UIs into place and are very easy to use. Thus, it's a matter of a few days or weeks to learn a new tool.

# An overview of an end-to-end data engineering process

57:29

Alexey

**There is one question and probably the last one, because I see we should be wrapping up. This question has seven upvotes, so I think I have to ask it. A question from Akshay is, “I want to know the real world end-to-end process of how a data engineering team designs, builds, deploys, and monitors structured data in both real-time and batch.” [laughs] I think this is too much for one minute that we have left, but maybe you can give us an overview?**

57:56

Rahul

I think we should have another session dedicated to this piece which can last a couple of hours, right? [laughs] But anyway, just to summarize this in a matter of a few minutes – there are different pieces of end-to-end data engineering. First, you start with the source systems – where data exists. It could be a relational database, flat files, cloud, S3 – all these things. You build one data pipeline to pull data from the source systems and load it into your data warehouse or data lake, which is a centralized data hub.

Then you build one process to expose this data. This is the first part of it, you have all the data in a data warehouse, but how can a consumer, who is interested in your data, consume it? There could be various methods of consuming the data – you have a reporting tool or visualization tool. You also have certain API's built on top of your data warehouse platforms, which can serve any applications in real-time. You can have another ETL layer for some other consumers who are interested in our data. So, this whole end-to-end system can be in batch mode as well, which runs daily or weekly loads.

It can also be event-based as well. The moment the data arrives in a source system, you can have certain events listening to that source system. For example, let’s say you have an S3 bucket. Your file lands on an S3 bucket, your lambda function gets triggered. I am talking in terms of AWS terminologies. That puts your reads data from S3 buckets and loads into your data warehouse. That is more real-time. Or if you talk about streaming, then you have Kafka or any message brokers, which can connect to real-time data with your warehouse. Just to summarize this focus, it’s not only about just bringing data into the data warehouse, but there’s a second part, which is how you expose that data. If nobody is able to consume your data, then your data warehouse is not valuable or useful at all.

# Finding Rahul online

59:59

Alexey

**Yeah, thanks a lot. What's the best way to find you if somebody has questions? Is it LinkedIn or are there other ways?**

60:08

Rahul

Yeah, LinkedIn. These days, I’m spending a lot of time on LinkedIn. So yeah – please reach out on LinkedIn. You can maybe put the LinkedIn profile in this description.

60:20

Alexey

**I will. And if I need to put anything else in the description just send me the links, and I will do this. I would like to thank you a lot for joining us today, for sharing your experience, and for answering so many tough questions. Also thank you for sharing your personal experience of transitioning and telling us about the problems you faced. I would also like to thank everyone who joined us today and asked questions. I would like to wish everyone a great weekend. We’ll see you next time.**

60:50

Rahul

Thanks for inviting me, Alexey. It was a really interesting discussion. I did not realize that one hour had already passed. Thanks for asking great questions, guys. Stay in touch. You can find me on LinkedIn, as I mentioned. Hope to talk to you guys again.