1:45

Alexey

**This week, we'll talk about the journey from being a data manager and transitioning to a data architect. We will discuss it from both the technical and leadership perspectives. Today, we have a very special guest, Loïc. This is not the first time you see Loïc here. Previously, he gave a talk about building a data lakehouse. He mentioned many interesting things like being a data architect and being the Data Manager. During that talk, we thought that it would be a really awesome idea to have Loïc again on our show but as a podcast guest, where we can talk in greater detail about his career, what he's doing, what he was doing and how he made the transition.**

**Loïc is the data lead at MyLight 150. That's the name of the company, right? He'll probably tell us more about what he does there. He has more than 10 years of experience in the data space in various roles from being a data manager, doing database management, doing data engineering, being a product owner, being a tech lead, being a data architect, and being a data lead. A lot of stuff. Thanks a lot for finding the time to join us today for this interview, Loïc. Let's start.**

# Loïc's background

3:15

Alexey

**Before we go into our main topic, let's begin with your background. Can you tell us about your career journey so far?**

3:24

Loïc

Yeah. In a nutshell, as we say, I started as a data manager back in 2013, working in the UK in a company called SIXENSE by now. I was doing a lot of data management, gathering data and just making reporting available for other people (stakeholders) mainly dealing with civil engineers and the civil engineering industry in general – the construction industry. Then I became interested in the data engineering side of things just because I was maybe frustrated a little bit with gathering all the data together, mix and matching CSVs, XMLM – whatever the format it could be together. I became a data engineer – it was a really technical part of data engineering, like ETL, groupdash and so on.

Then as time goes by, you focus more on what other people require in terms of data and needs – what their needs are – so I took positions as a data consultant at CGI back in France. Then I was a project owner (technical lead) in a sort of data factory team. And right now I'm the data lead at MyLight 150 What do I do there? It's a mix of two hats, maybe a bit more. The first one is very technical – we put together (as you may see it in the previous talk we gave on the lake house) this was a very operational side of things, let's say. The second hat is about the management of the team. I guess there were a series of questions just to transition through the career and through those really technical parts to somehow more of a leadership role. So that's it with my background, I guess.

# Data management

5:44

Alexey

**When I heard the term “data management,” what I had in mind was more like managing a team, so what you do now. But from what you described, I understood that it's a different role. It's more like an analyst role in a non-tech company, where you would get a lot of different data from different data sources, and your job was to prepare reports.**

6:14

Loïc

Yeah, it was exactly this. My customers were basically civil engineers – they required a lot of data to analyze the good health of their structure (by the way, the practice is called structural health monitoring, where you look at the structure and you check for cracks that advance and so on) and my goal was just to take data from a lot of sensors that we were installing in the city and buildings, and just mix and match them together to provide trends and analysis.

In today's nomenclature of data roles, this looks more like a data analyst role, but at that time, it was called data management. Because it involved a little bit more than just data analysis, like putting the data together and saving it, and sharing it. Also, there was a lot of the data discovery side. So that's why it was called data management at the time.

7:16

Alexey

**So it's like a combination of data analyst and data engineer.**

7:21

Loïc

Yeah, but more of the data analyst side, actually. Then I transitioned to data engineering, because I was spending days – hours of my days – just mixing and matching data. Also, I have a civil engineering degree. Nothing to do with data or IT or whatever, but I was always taking the computer science-oriented specialties in school. I was very aware of how to program and how to deal with databases and so on. Naturally, I wanted to automate my work of binding all of this data together. I remember I was spending probably eight to ten, twelve hours a day mixing and matching data. In the end, I was taking like one two hours just to automate a little bit. Then the next day, for one or two hours, you automate another bit of it.

In the end, you work extra hours at the beginning, but you have a fully automated process a few weeks later. This was what basically got me into data engineering – automating data analysis work got me into data engineering. It was, “Okay, if I automate the end of the process, maybe it's going to be interesting to automate the data source to the consumption of the data.” I really enjoyed automating the end result, so I was like, “Okay, I could basically do this for a living – automating data processes.” So this is how I got into data engineering.

# Loïc's transition to data engineer

9:21

Alexey

**So you became a data engineer within the same company, right?**

9:25

Loïc

Yeah, exactly. Data management at SIXENSE was very broad in terms of practice. We would install IoT devices and you were to configure the data loggers, for example, to reach the data from sites, the ETL process to load it into the database, and then the reporting process as well – from the database to the software solution we were providing. All of this required everything to maintain the full data pipeline. There were a lot of different aspects to the data manager position – it was a very broad role. But you could do most data analysis, a bit of data engineering, and sometimes, when you want to do statistics on your data to understand the trends and so on, it was data analysis/a bit of data scientist to understand what's causing the trend or whatever. So it was a very broad role – data management. I really liked the data engineering side of this, which when I think about it, was maybe linked to the fact that I really liked software programming. This was the part that I could have more control over. Maybe this is just me liking the controlling side of things.

10:59

Alexey

**But also, as you mentioned, you have a degree in civil engineering. So for you, all this data that was captured on sensors about cracks, sediments, and all this stuff – you could really make sense from this data, right? You could make sense of this data, you could understand what's happening there, you had all of this domain knowledge. Did it actually help you with the transition from being a data manager to data engineering?**

11:27

Loïc

What helped with transition was me spending a lot of my spare time learning about software engineering good practices, reading about all the database management and CRUD operations and all of this – more investing my personal time on learning the data engineering things. But it is true that as I knew who was going to consume the data that we were producing, it was a big plus because I knew when something was wrong, I could diagnose where the data quality problem started to happen or whatever. In the end, this first job as a data manager transitioning to data engineering was really the perfect job because it was in the civil engineering industry – I was reporting to civil engineer and construction managers. But I got two feet inside the data space straight away.

It was a perfect transition from my degree. We are talking about the transition from data manager to data architect, of course – data engineering. But the transition was actually a civil engineering background to the data space. If you look at the market today, a lot of people are switching careers and usually, they do it after a few years in the work – like they work in finance for a few years and then they realize, “Okay, what I really liked was the data,” whatever. But in my case, straight after my diploma (graduation) I knew I wanted to do something else other than civil engineering. And it's very fun because at the school where I studied, there were always these charts about what the people in the school are becoming like 10 years down the road and there was this category of 15–20% of the students that work in a field that is in no way related to what they studied. So I am in this category right now. [chuckles]

13:57

Alexey

**[chuckles] Yeah. Life is interesting, right? You never know. You graduate from school and you have no idea what you'll do in 10 years, because it's really difficult to know. [Loïc agrees] Parents say, “Become a civil engineer because it's a well-paid job.” Then you go there, you study, only to find out that you like other things more. But it's a journey. Right?**

14:22

Loïc

Exactly. Sometimes when you look for a new job or you are in the interviewing process, you always have this classical HR question coming in, “Where do you see yourself five years down the line? Ten years down the line?

14:37

Alexey

**I wish I knew. [chuckles]**

14:38

Loïc

I wish I knew. [chuckles] My discipline did not really exist 10 years ago – I assumed I was destined to be a civil engineer. So who knows? [laughs]

# Challenges in the transition to data engineering

14:51

Alexey

**For you, what was the most difficult part when you did the transition? I'm guessing that as a civil engineer, you did not study software engineering and you needed to invest a lot of time in learning all these data engineering fundamentals. [Loïc agrees] Apart from that, what were the most significant challenges that you faced there and how did you overcome them?**

15:19

Loïc

Yeah. The thing is, during the last decade, a lot of things happened in the data space. As you know, IoT became a thing – it produced a lot of data. I was in the IoT/civil engineering industry, so I can testify to that. Of course, you had the classical business intelligence practice, which was somewhat well-established – people doing the SSIS packages, building data pipelines, databases, reporting about the profitability margin, whatever.

15:57

Alexey

**SSIS is this tool from Microsoft, this is the integration service where you drag and drop things – you connect different squares with the mouse, and then this thing somehow works at the end. Right?**

16:09

Loïc

Somehow it works, because you do your high-level data pipeline in a sort of “low code/no code fashion,” but then you still need to call stop procedures, and you still need to build those SQL, TSQL – you need to do that anyway. You just have a nice interface to architect your code somehow. Yeah, it's built into the...

16:34

Alexey

**It's been a while since I saw this thing.**

16:37

Loïc

Yeah, it's been a while for me as well. Right now, it's Data Factory, Airflow, whatever – those things are basically SSIS legacy. So we have these these days, and everything is cloud-based. So yeah, the IoT, the volume of data, the fact that cloud was somewhat booming as well – if I remember, the transition happened around 2010 for us. I'm sure if you Google something like “Azure adoption graphs,” or whatever, you will probably see a big spike in 2017–2020 – this is when a lot of companies just shifted to the cloud – this happened. In addition to the big volume of data, this became somewhat of a new Wild West, because you want those new cloud services and a lot of data to deal with.

Your classic documentation or resources, like “How to build an SSIS package (or whatever),” was not really relevant anymore. You had to find all the communities and so on. This was a challenge, actually – to transition from a place where the volume is increasing and people are shifting from cloud to cloud. You arrive into this new space where you have everything to rebuild somehow, but you have the good practices and concepts, and you need to have some basics, if I may. The best thing to overcome this challenge was to just keep yourself updated on how people do things – usually, Stack Overflow was filled with questions about people having problems with those new technologies (with those new services). A lot of communities have been built around those platforms. Or in my case, I took a few notes, just to make sure I didn't forget anything. I tried a lot of scripting languages – I ended up using Python because it was the most used in the industry. I didn't think more about it than that, “It's being used, I'm gonna use this. Okay.” Same for the cloud.

You had AWS at the time – Google Cloud was somewhat there and Azure was just investing a lot in it. My company was using Azure Cloud, of course. In a lot of job postings, they were recruiting people using Python and cloud. For me, it was a very data-driven and practical choice to just go for it. But in the end, as a feedback, all the clouds are really more similar than dissimilar. What you will find in the platform will look mostly the same in another cloud provider. I would say not to focus on getting certifications to prove that you know the cloud or whatever. Because what's most important is to grasp the very strong basics of “What is a CRUD operation? What are the types of services you have access to store your data? What are your options to build? What are your tools available on this platform?” If you know that you need a hammer to just insert a nail into a wood plank – you will find your wood plank, you will find your nail, you will find your hammer in this new platform. No worries. This is what I would advise.

20:48

Alexey

**That's advice you share as a data manager right now? I mean, as a data engineering manager – because you're hiring data engineers currently, right?**

20:56

Loïc

Say it again?

20:57

Alexey

**Do you hire data engineers currently at your current role?**

21:01

Loïc

Yeah, I actually have one data engineer who has more experience than myself on a lot of the BI and data architecture side of things. But still, when it comes to Big Data and using Spark and new platforms like Databricks and whatever, there is still a transition to do. When I hired [people], the interview was basically more focused on, “Do you have some skills about doing data pipeline projects?” and “Tell me how it's going to go wrong. And what did you do?” and somewhat challenging strong opinions on good practices vs reality, etc. So when I hire for a data engineering role, I don't look for the perfect certifications – I look for experience, and I look for the scale of the projects people have been working on, the scale of the teams that they have been working on, and the general attitude towards solving the problem with the tools you have. If people are afraid, for example, when I say “Yeah, we are on Azure Cloud,” if they are afraid or they are like, “I only know AWS,” for me, it's somewhat of a red flag, because I would prefer an answer like, “The clouds are more similar than dissimilar. I know AWS, but I will adapt.” This is just about the mindset of answering the question more than the answer itself.

# What is a data architect?

22:47

Alexey

**Okay. Fair enough. I just looked at the time and I see that we spent most of the time talking about data engineering – your transition from data manager to data engineer. But we also wanted to talk about your other transition – the transition you made from a data engineer to a data architect. Before we talk about the details of your transition, I was wondering what a data architect actually is. Who actually is a data architect? What do they do? What kind of responsibilities do they have?**

23:21

Loïc

Yeah, that's a very good question. Actually, I had a couple of students a few weeks ago interviewing me just on this particular question, because they had to fill a form about, “What is a data architect?” to have this new position in school. The first bit of the answer was that a data architect is not a junior position. You do not graduate as a data architect. This is a role that you acquire when you have been firmly walking into the different aisles of data management from end to end. This was the first bit – it's an experienced role, because you need to be aware that you need to architect the data from the data source to the staging area, to your data warehousing part, and then building datamarts for the people to consume the data.

At each stage, you need to understand, for example, how the data is being produced and how the data is being consumed. Most of the time, there are automated systems like IoT producing data. In that case, it's actually the easiest part of data engineering. But when there are people producing data. You need to understand the processes. Then, when there are people consuming the data, you also need to understand the use case and what the final result of the analysis is and how they are binding the data and so on. I think the data architect role is about bridging the gap between the ETL CRUD operations (very technical) and the people using the data (producing and consuming it). There are a lot of definitions of data architect, but what is important is that it's not a junior position – you need to have experience on the full chain – and that you need to focus on the processes and the people and the use cases of the data more than the technical side of things. That will be the two main points, I would say, that describe a data architect.

You need to know about modeling of the data, because people are going to use the data that you have somehow created (pre-prepared) for them, so you need to make sure that your technical process of extracting, collecting, managing, and modeling data matches – that it's good technically, but it's also good business-wise.

26:11

Alexey

**Okay, so it's about bridging the gap, as you said, between the requirements and the implementation, right?**

26:19

Loïc

Exactly. Yeah.

# The output of a data architect's work

26:21

Alexey

**And for that, you need to understand the processes, use cases, what the final results should be, more or less. It's a very technical role, so you need to have experience doing things end-to-end – from source staging, warehousing, datamart – all these things need to make sense. Finally, you mentioned that you need to have a good understanding of modeling data. [Loïc agrees] How exactly the data looks like in the staging area, how exactly the data looks in the warehouse, how the data looks like in the datamart. But at the end of the day...**

**Okay, you spent a lot of time talking to different stakeholders, you spent a lot of time talking to engineers who are going to implement that, but what is it that you do? What is the main, let's say, outcome? Is it a document that describes that, “At this step, you do that,” there are some diagrams with arrows showing how the data flows or is it something else?**

27:20

Loïc

Yeah. What is the output of my work? The most important thing is, I would say, team alignment. Because when you have a data project, it's not only one team, creating the data, managing it, saving it, analyzing it – this would be the perfect scenario, of course. But usually, you have a team that is creating data, another team analyzing the data, and another team processing the data. The outcome of good data architecture, I would say, is that that process is optimized, of course, but this is only the technical part. The most important output of a data architect is team alignment, when it comes to producing data in a way that is usable by the pipeline, and storing the data in a way that is then usable for the business.

Of course, you will have a lot of tools to help you do this. Most of the time, as I showed in the previous talk, it looks like a massive bowl of spaghetti, with all of your data processes and all the flows of data from A to B to C etc. But these are only tools – the most important outcome is to have the alignment of the teams that are producing, extracting, transforming, and consuming data. This is the number one output of a data architect, I would say – making sure it's smooth.

29:03

Alexey

**I guess, in order to have this alignment, there needs to be some documentation. There needs to be some written piece or something like this “bowl of spaghetti” diagram, as you said. There should be something... Not physical, but something in your documentation that describes “Okay, these are the requirements. These are the limitations. These are the stakeholders. These are the users. These are the requirements and this is how you will want to implement this.” And all these teams, like the team that creates data, the team that processes data, the team that analyzes data, all have access to this document and they say, “Okay, this is what we want and need.” Right?**

29:48

Loïc

Yeah. Usually, if you have a dream project where you live in theory land...

29:55

Alexey

**Dream project, okay. [chuckles]**

29:56

Loïc

This is the perfect thing to do. You can spend a lot of time doing a really detailed a specification about all the processes, “It will go like this – from this source, connected with this protocol, you will get the data and you will store it in this way, into these tables, and that way, with those columns, and blah, blah, blah.” But reality has shown that whatever you plan, don't plan in too much detail, because a lot of things are going to derail you from your plans. The most important thing is to have common good practices and concepts, like “What is the quality level of data expected to arrive at the bronze level? You can't accept a lot of null values or things like this. And then if you ingest for the first time, you realize with a quick analysis that, “This is garbage.” You probably have a data process to improve, like the way people are entering data in forms or whatever, so that they do not skip the question or they do not choose the default null value – whatever. You make sure that your application – your UI or whatever – helps you to make all the data [inaudible].

So this is more about common sense and good practices. Make sure you have good data arriving at your bronze level and then understand how people are going to use it so that you say, “Okay, you will need to analyze those analytical metrics.” You want to analyze data stores, for example. You have the name of the store, the region where it is located, the period where the sales happened, the margin of the sales, and so on. All of this gives you your dimensions of analysis, like geography, time, stores, the article (the thing you are selling, for example). Then you have your metrics, which are the turnover, margin, number of sales – you need to basically build... [cross-talk]

# Establishing metrics and dimensions

32:26

Alexey

**But stakeholders give you this information. They say, “Okay, we care about these things.” Right? At the end, when we analyze the data, “This is what we want to see in the dashboard. We want to see these kinds of metrics.”They tell you that and then you need to define, talk to other teams who create data, who analyze data and process data, and you work this out – you need to understand what kind of dimensions there are, what kind of metrics there are, all these bronze things you mentioned. Bronze is the staging area, right? Or what is that?**

32:58

Loïc

Yeah, bronze is the very raw data, and then you have silver and gold. What you are discussing with the stakeholders, who are going to consume the data, is more like, “What should the gold layer look like so I can analyze it?” And then what you are discussing with the people that are producing data, it's more about, “What should be the acceptable quality level that you can drop (dump) into the bronze layer.” Then, together with your data engineering team and the analysis team, you discuss, “Okay, I got this, I need that – how do I mix and match? How do I bind it? How do I transform it so it's the appropriate level of quality for my analysis?” So the most important thing is that you need to discuss that with people.

Generally, stakeholders don't say, “I have an analytical dimension that is the geography and I have a metric that is my turnover.” Never. Okay, maybe with very lucky people you will have those discussions. [chuckles] But usually it goes like, “I need to analyze the margin in this region.” Nowhere in the sentence, “I need to analyze the margin in this region of the world.” Nowhere in the sentence did they mention the dimensions and the metrics. You need to discuss and say, “Okay, your metric is the margin and your dimension is the geography.” Maybe for this analysis, if I manage to store my data in the proper way, you could be able to scale this analysis and have your margin in all the regions of the world. So your analysis is not only for this area that you want right now – this is your quick win, what you need to output to your CEO for next week – but maybe what you want is a more scalable process to be able to reproduce your analysis very quickly for another region, at another time, whatever.

This is the kind of discussion you have with your stakeholders, just to make sure that you are going to store the data in the proper way. You will have your facts table and then you will have your geographical dimension and your store dimension and your article dimension – and your fact table will have all of those columns and help you to build those metrics and so on. This is where the technical side kicks in, but the role of the data architect is really understanding what to build and why. This is the key role.

35:43

Alexey

**How exactly they will use it once it's built – what kind of questions they want to have answers for. Right? What kind of analysis, how exactly they will use it, and also maybe what sort of decisions they will make based on whatever they want. Right?**

36:00

Loïc

Yeah, exactly. Usually what happens in a company is you have different departments – in our case, for example, we will have Supply Chain, Finance and Sales – they will also analyze the quantity of stock we have left, but for very different reasons. And you will end up building a report for the sales, another for the finance, another for the supply chain – but still, the data that you use at the origin is the same for all of them.

As a data architect, you need to also be aware of this so that you can put together all of those developments to build a sort of core model or some foundation to build all of those different use cases. So it's not only from source to consumer, but you also will have a lot of consumers and there is another dimension that is the transversal dimension of the work, if that makes sense.

# The importance of communication

37:10

Alexey

**To me, it sounds like this work mostly involves communication, right? You need to, first of all, speak with stakeholders, understand what the requirements are, how it's going to be used, and then you need to spend a lot of time talking to teams, to understand maybe what is the current status, how this can be implemented. Then you need to come up with this design document, to make sure that all these teams are aligned. Right?**

**So would do you say that as a data architect, you spend most of your time talking with other people? Or what's the breakdown? What exactly does it look like? What do you say it's 80% communication, 12% documentation writing – do you do any hands-on stuff? What does a typical day look like?**

37:57

Loïc

It really depends on your company. If I talk for myself, the previous year has been really focused on the technical side of things. I was mostly hands-on, somehow building the platform with the rest of the team – making sure all that was flowing from the sources to the data lakehouse. Right now, we are in the process where the team knows how to do all of this – we have a common set of practices. The next stage is to finally have time to discuss with the stakeholders about what it is that they need and build a better gold layer, I would say – focus on the gold layer of the data warehouse. So bronze and silver are solved, and now we are focusing on the gold. The role has been shifting from something like 80% of dealing with technical people and maybe 20% of dealing with stakeholders, to the actual opposite where, once you have the data, you can just spend more time focusing on discussing what you want. So it's going to be 80% stakeholder, 20% technical, for example.

So it really depends on the phase of the project where you are, I would say. But my advice is that stakeholders really want to see that you are progressing on the project – on the data they want. So you should focus more on the stakeholders. As a priority, you should focus more on stakeholder management than your team management once the good practices have been set up and agreed within your team. So the first thing is to agree on the good practices, build an end-to-end use case, say, “Okay, this is how we do this. This is all standard.” And then you can free time for yourself to deal with the stakeholders. Right now this is where I'm moving into, if I talk for myself.

40:18

Alexey

**From what I understood of what you said, what you do is quite a technical role. So you are still pretty hands-on with building things with the rest of the team. But still, there is this component of talking with stakeholders, and it's your job as a data architect to actually inform them on what the progress is, which stage it is currently in, and of course, if there are any questions (if something is not clear) it's your job to approach the stakeholders and clarify the requirements.**

**For example, “When you said that you need to analyze margins in some regions of the world, did you mean on the country level or did you maybe mean county level?” So you need to go to them and ask that. Right?**

41:07

Loïc

Yeah. I don't believe that if... I am present in all the meetings where we specify things the data team can scale, basically. So the transition is about somehow empowering your team so that once they know the good practices and how to do things, that they do it for themselves – the data analyst going to the business, discussing what they need, and then discussing with the data engineer. The data engineering parties are going to know how to do things in an optimized fashion, and then at least are going to know what the business requires.

I think that part of the role of a data architect is to help both with practices and to communicate well together – to make sure that the communication is smooth between the data engineer and the data analyst. Also, that the communication is smooth between the data analyst and the stakeholders. The end goal looks more like auditor/data product manager, if I use “techie” terms.

# Setting up best practices for the team

42:31

Alexey

**So it looks like you want to teach people how to speak, so you can kind of get out of the team – so that they work without you. Right? What I mean by this is, if you're involved in everything, then you become the bottleneck and nothing is moving. So what you want to do is set up these best practices so that they know how to talk, who to talk to, and when to talk to them. Then, you can step away and watch this thing work without your participation. Right?**

43:03

Loïc

Yes. Stepping away and just watching it go. It's a new level, when you know what the company is going to look like in three months, six months, whatever. You need to make sure that the effort of your teams, in terms of priority, that the time they are spending doing things is aligned with the short-, mid-, and long-term objectives of your company. I think this is also a part of the data product manager or data architect role, depending on the size or the topology of your team. But the most important thing is that you know how to do technical things, you make sure the communication is smooth, and then that you empower your teams so that they know how to do things individually so you can transversely scale your work. Then you can just focus on prioritizing the work of the team so that it aligns with the company's objectives. Does that make sense?

44:11

Alexey

**It does, yeah.**

44:12

Loïc

Good.

# Staying relevant and tech-watching

44:13

Alexey

**I see a question about handling – how you adapt and stay relevant in the field. In my experience – so I was kind of also working in an architectural role, even though it wasn't related to data engineering, it was related more to machine learning, but that doesn't matter. What I noticed is that the more I spend time working with stakeholders, communicating, and aligning teams, the less time I have on the coding side (on the technical side). What happened is, after half a year of me doing this thing, I became very not hands-on. I stopped coding.**

**I spoke with other people that work as principal engineers, or architects – all of them confirmed that they went through something similar. It was a similar process, where they started doing more high-level stuff and with time, they became less hands-on. With time, new tools come out – DBT exists now and it's very popular (five years ago, it wasn't there) and, if your data engineering experience comes from five years ago, you might not even know how to use DBT. You just know that it exists.**

**It's the same with machine learning – these new tools keep appearing, there's some buzz about them, but if you lose touch with the ground (if you become too high level) you risk becoming irrelevant. You risk forgetting things. So how do you do that? In your case, you said you still try to be hands-on, but my question is – how do you find time to do all that?**

45:57

Loïc

It's actually very funny that you mentioned DBT, actually, because this is exactly what is happening to the data stack in the data team and myself At MyLight. We are working on this gold data level, and I have this new data engineer in the team, who is a very good individual contributor – this is what he likes and is meant to do. So, speaking about DBT – he comes from a very SQL-based background and the data stack we had was not really manageable in terms of the gold level. DBT is perfect for this area, where it's automatically building your asset diagram, and your pipeline, and scheduling everything for you, and so on. But, as I was spending a lot of time doing specifications or stakeholder management, I couldn't install DBT myself – set up the pipeline, build all of those Jinja routines, and so on, etc.

Actually, I did not touch DBT at all. The data engineer was the only one dealing with this. But we were doing one-on-ones – every week, I have 30-minute meetings with all the team members. And during those one-on-ones, there is, of course, the synchronization of the work that has been done, etc., etc. But part of it was, “Okay. Tell me what you have done and go into detail, because I want to see what you did and I want to understand it.” Of course, you have other high level things to do, but those one-on-ones, when you are managing a team of very technical individual contributors – they are your perfect occasions to stay relevant on the technologies or tools that you are implementing on your data stack, for example. This is the first place where you learn, ask questions, try to show your screen, you show the code, you show how it works, you run the pipeline end-to-end, etc., etc. Of course, you are not doing it hands-on, but you know that if some shit happens, basically, that you will be able to be a second set of eyes on the work of the person, and be able to debug the pipeline together or whatever.

You can also help the team on getting good practice with new tools, even though you have not implemented them yourself – you just share the knowledge about what you did in the past, what works and what doesn't, so the team is then somehow empowered as we said and just make the decision for themselves. But this is a very good way to stay up to date and relevant – those one-on-ones with your individual technical contributors. The second thing is to stay up to date – you read blogs, you subscribe to very good seminars and podcasts, like DataTalks or any other that you like (that is in your field).

Stay up to date by watching webinars, and when you have something that sparks your attention or excitement, you maybe spend a couple of hours during your workweek or your evening just trying to make it work for a little project or some part of your project, as a proof of concept. Build a proof of concept that works, that will include this new technology, and see if it's worth it. If it's not working – good. You have the hands-on experience now (maybe) and you tried it, you benchmarked it, and it was no good. But for you, there's the next thing. At least you are aware of what is there and how to use it.

50:23

Alexey

**Basically, if I tried to summarize what you said, it's to look at trends – what exactly is hot and what people are talking about. Secondly, try to squeeze in some time in your week (in your calendar) to find out where you can actually get some of these tools that people talk about and then implement some sort of proof of concept. Right?**

50:45

Loïc

Yeah, “technology watching”. This is the term. Create yourself a community on LinkedIn so you can be aware of [these technologies]. Find an expert about Power BI, find an expert about data engineering, find a newsletter about data science, find a DataTalks [podcast] about machine learning – whatever. Just add a lot of streams of input so that you can keep aware of the trends, of the new things. Have a lot – not only one – have a lot of them so you can see things that are repeating and you can see the trends. You see, “Okay, this one talked about this technology – DBT DBT, DBT.” If everybody's talking about DBT, you should be trying it – otherwise, you are missing out on an industry trend, which could be bad for you in the long run.

51:38

Alexey

**So this is how I personally missed out on all this GPT stuff. [chuckles] So now I have no idea how it works. It feels like all the data scientists know what exactly is happening with all these LLMs, but I have no idea. I can just use them. It's the same with DBT, though – it's similar.**

51:57

Loïc

That's okay. GPT is quite fascinating. A lot of people are just diving into the hype of GPT. Good for them. But I think we are in a very busy world. At some point, you will find the time to just take a deep breath, go out of the water, and just have a look at this GPT thing. By then, everything will have somehow consolidated – this is good GPT stuff, this is bad GPT stuff, this is half-good GPT stuff, or whatever. And you will have more insights than just trying everything straight away as it comes out. You will lose a lot of energy if you focus on GPT right now. Just know what you can do, keep thinking about what it can do and at some point, just consolidate everything – do a deep dive at that time, I guess, when it's somewhat mature. At the speed at which things are moving these days, its maturity is a very obsolete concept. [chuckles]

53:09

Alexey

**[chuckles] The problem is, apart from GPT, there are so many other things that are also trending. You open Twitter or LinkedIn and people talk about all these other things. Then it's like, “Okay, how do I find an extra 24 hours in my day?” Right?**

# Setting up specifications for a pipeline

53:28

Alexey

**We have a few questions. Mohammed is asking, “How do you manage data specifications while setting up a data architecture pipeline for a project? Is it something you do in parallel? Or do you first come up with data specs and then a pipeline, or first the pipeline then specs? What does it look like?”**

53:48

Loïc

I live on a planet near a black hole. When I spend one hour here, it's like seven years on Earth. You know? [chuckles]

53:58

Alexey

**[chuckles] That's convenient.**

53:59

Loïc

It's really complicated. I touched on this a bit earlier, but the only way to escape the fact that you can't clone yourself, or you can't work 48 hours a day is to scale your knowledge by giving it to your team. Then, you can do a bit of both. Once the work you have done in terms of specifications is mature enough, introduce a data analyst and a data engineer and then start the work.

54:36

Alexey

**So first the specs and then pipeline? I guess it's a draft of the specs, right? You don't want to have a super-detailed specification where everything is perfect, but when you try to implement it, nothing works. Right?**

54:51

Loïc

Well this does not work if you do a waterfall project, where you do a technical design – a detailed technical specification and so on – and what you will build will not be relevant when you will have built it. So just try to sketch end-to-end what it should look like, try to get the customers' feedback (your stakeholders' feedback) as soon as possible so you know if you've been doing something wrong and you have some insights from the domain experts. They will tell you, “Yeah, but look at this. The turnover is not good at all on this. You should not take into account those kinds of articles because they’re transportation costs. This is nothing that we include in the codes that we send to our clients.”

Anyway, you can get those domain insights and expertise knowledge very fast if you just quickly draft end-to-end and then you refine it with them, you bring more data, you specify, and you iterate (this is the most important concept). We are, I think, in a world that is... I don't know if it's largely agile, but in the tech industry, at least, we push those agility concepts more and more. Do not over-specify and do not over-optimize your code, when you are doing things straight away, but do try to get your clients' feedback straight away so you can iterate really quickly and get the 80% of the results in the fastest way possible.

# Be agile, create a POC, iterate ASAP, and build reusable templates

56:39

Alexey

**In summary, you draft an end-to-end specification (very drafty), you get feedback on that, then you incorporate the feedback, and implement a POC pipeline. Then you get feedback again because this POC probably produces something – get feedback, incorporate that, perhaps, into the specification (adapt the specification), change the POC and repeat. Right? At some point, the POC becomes a proper project where you fix all the technical debt. But this shouldn't happen immediately, right?**

57:12

Loïc

Yeah. Usually, this is a trap. You're going to build a book, and you will end up with a lot of technical debt, and a big impact on your code so you can make it scalable. If you have previously built projects, you have some kinds of templates on how to do things, like an ingestion template, a transformation template, a creation of datamarts template, Power BI templates with the appropriate colors and whatever. Make sure you template things that have shown results at scale in other areas of your projects. On other projects, make sure you reuse what you know how to do, because then you just gonna go fast.

Do you need a new source? Okay, I got my templates from the data source ingestion – boom. API? Okay. API ingestion function – I get it straight into bronze, merge into silver. Then, okay, they need geographical dimension? Okay, I know that I have my postal code or my country database somewhere – I'm just going to reuse this for the proof of concept. Then, as you are reusing bits and pieces that you've built in previous projects, these are the things that are already used in production by other projects, so you have a basis of elements that are reusable. You go from proof of concept, which is somehow already scalable, and modulate the differences and the specificities of your proof of concept. But you will go fast into the ingestion phase if you reuse templates. Build templates!

58:57

Alexey

**So it's also part of your job as a data architect to know which templates are already there (which templates exist), which templates you need to build, update, etc., right? You need to think about the things you've built in terms of templates, like, “Okay, like I see multiple projects and this thing here kind of repeats. There's some redundancy, so let's make a template out of this.” [Loïc agrees] As a Data Architect, you need to watch out for this. You need to look for this.**

59:34

Loïc

Yes, you need to be aware of templates. You have your data engineering hat, where you know that you have this somewhat software engineering side of you that tells you, “Okay, if I'm repeating something three times, I may want to build something common for that, which I'm going to reuse. Ingest from bronze? This is my function. I will put it somewhere and then I will use it in this project, in this project, in this project. I will call the same codebase that I will reuse into the different projects.” The data architect and the data engineer and the software engineer side of the personality that you are should be aware that you can't build the perfect generic function to solve it all. Otherwise, we'd be out of work a long time ago. But you need to balance what is generic and reusable and scalable with what is specific to every project. You can't build a key that is going to open all of the doors in the world, but 80% of the doors is already good enough. We'll open the remaining 20% with my specific keys that we will build.

# Reaching out to Loïc for questions

60:51

Alexey

**You'll pick the locks, right? [chuckles] [Loïc laughs] We should be wrapping up and I see that we still have quite a few unanswered questions. Would it be okay if Mohammed and other people would reach out to you on LinkedIn with these questions?**

61:08

Loïc

Sure, sure, sure. The link will be in the description, as you will say. [chuckles]

61:16

Alexey

**Yes, exactly. [chuckles]**

61:17

Loïc

There will be my LinkedIn profile in the description. If you have questions – I didn't answer them because I'm very verbose when I speak – reach out. It will be a pleasure to answer.

61:31

Alexey

**Thanks, Loïc! It was, as always, great to talk to you. Time flies – we are already over time. It was a really big pleasure to speak with you again, so maybe we should repeat. We actually... The funny thing is, we did not discuss what exactly you did for the transition – we mostly talked about your transition from data management to data engineering, and then we talked about the role of the data architect. But we kind of missed the actual transition. [chuckles] But, for me, it was interesting. I hope it was also interesting for everyone else. Thanks again for your time. Thanks, everyone, for joining us today and have a great week.**

62:12

Loïc

Right, thanks. See ya.