1:14

Alexey

**This week, we'll talk about transitioning from data science to data engineering. We have a special guest today, Ellen. Ellen is the head of data engineering at WhereIsMyTransport, which is a company that provides mobility and location data for emerging markets. She has been working in software engineering, data science, and data engineering roles for over a decade. A common theme across her career is her passion for building high-quality technology of which data is a core component. She also enjoys teaching, speaking, and writing about data topics. Welcome, Ellen.**

1:48

Ellen

Thank you. I'm happy to be here.

# Ellen’s background

1:51

Alexey

**Happy to have you on. Before we go into our main topic of transitioning to data engineering, let's start with your background. Can you tell us about your career journey so far?**

2:01

Ellen

Of course I can. I studied computer science at Uni. It was my first degree and I really enjoyed it. I specialized in software engineering and what's called in German “Wirtschaftsinformatik” which translates to business applications of computer science. Then I worked as a software engineer for a bit in different countries, but then I got really bored with backend engineering because I felt I was doing and building the same kind of “read data from the database, put it into an API – get data from an API and put it back into the database” thing over and over again. At the same time, I started a psychology part-time degree. I didn't really enjoy the psychology part that much, but I was really surprised that I really enjoyed a statistics course, which is obligatory for psychologists.

At that time, that was when data science became a bit of a hype topic and I thought, “Hey, maybe that's a cool way to combine my interest in technology and statistics,” so I set out on a path to become a data scientist. I did a Udacity nanodegree, which is kind of an online bootcamp to level up in machine learning and a bunch of other online courses. Then I got a job in a data team and insights team at SoundCloud, where I had a pretty wide range of things that I did – everything from data pipelines and scheduling, but SoundCloud at the time had its own data scheduler. But also, I did a lot of predictive analytics stuff, and plain old dashboarding and everything that's in between that. That's why I officially own the title of data scientist or senior data scientist. Then SoundCloud had a massive round of layoffs, which was quite painful to me because I really enjoyed working there in 2017.

Then I worked as a data scientist in a few other companies, again, doing predictive analytics and dashboarding. I was also doing production-level modeling, so everything in between. But I realized that I didn't really enjoy data science that much, because it felt very blackboxy to me because I was like feeding some model but I didn't really understand what was going on, which is partially due to the fact that my mathematics background isn't that strong. So I didn't ever really get the theory behind it. But also, I realized that I kind of miss the engineering part of it more. That was also the time where data engineering became more routine, because companies realized that they need not just data scientists, but also people that actually come before them and make data available for data science. At the time it was a common frustration for data scientists to not have data. Great ambitions – but no data. Then I got an offer at Native Instruments to be one of the first data engineers and I did that for a while. I still had the ambition back then to transition back into data science, but I left Native Instruments for unrelated reasons **b**efore that happened.

Then I got another job at another company as a BI developer. My title was data scientist, but really, what I did was develop custom data visualizations. I eventually realized that this was kind of a dead end for me, so I remembered my software engineering background, which I truly enjoyed except for the part that it was so repetitive. At the time, I was speaking to somebody at ThoughtWorks. She was the first data scientist at ThoughtWorks Germany at the time and she told me that ThoughtWorks was planning to expand its data engineering offering and asked whether I would be interested in becoming ThoughtWorks Germany's first data engineer and to help expand that offering. I thought this was really exciting, especially since ThoughtWorks is a great place for really learning good software engineering practices.

As a data engineer it is always frustrating and as a data scientist that always frustrated me that the quality of the code and the collaboration wasn't as good as I remembered from my backend engineering days. So I jumped on that opportunity and that's when I finally decided to stay in data engineering, which I had done on and off before. After ThoughtWorks, I went to where I am now – at WhereIsMyTransport – and built up the data engineering there. This is where I am now heading a team.

# Why Ellen switched from data science to data engineering

6:32

Alexey

**That's quite a story, thanks for sharing. What I also wanted to ask you and the next question that I prepared is – you were in data science and then you switched to data engineering, and I wanted to ask “Why?” I think you partly answered that. You said that data science was a bit too blackboxy for you – you wanted to understand the theory behind this. I mean, at work, you don't really need to know the theory, right? You have an algorithm, give it some data that you get out of the model and then you spend most of the time just doing other things, so you don't really focus on the machine learning part.**

**You also mentioned that as a data scientist, the code that you were writing wasn't really high quality. This is something that maybe because it isn't the focus for data scientists. You focus on other things. Were there other reasons why it was clear for you, “Data science is not for me. I want to go more into data engineering.”?**

7:38

Ellen

Yeah. Those were two of the main reasons, but there was another reason as well. I realized that – at least at the time when I was trying to be a data scientist – if you didn't work in a large company, like maybe OLX or Zalando, or one of the really big giant companies, the roles of data scientists were pretty frustrating. I'd never wanted to work in such a large company. If you start up a data science department, you’re usually just stuck doing a lot of really drudgery work and I got tired of that. In a lot of companies, and that's the sad truth – engineering is more respected than data science, except for those that really focus on data science. So I realized that whenever I did more data engineering work, I had a more comfortable working environment, my skills were more in demand and I didn't have to fight so much for getting anything or proposing ideas, and things like that. So it was also a question of professional respect for me – I was more comfortable with the environment that I was experiencing as a data engineer rather than as a data scientist.

8:42

Alexey

**That's an interesting point. I didn't hear about this in Zalando, or OLX, or other companies here in Berlin. But I did hear that a lot from big tech companies, like FAANG – Amazon, Facebook (which is meta now), Google, and so on – that, as you said, engineers are more respected. Also I heard only anecdotal evidence of that. People tell me that engineers are like first-class citizens while data scientists aren't. I am quite surprised to hear that, to be honest. But I guess since many people mention it, this is becoming a pattern, which is pretty sad.**

9:32

Ellen

It is sad, yeah. I don't think it should be that way at all, but it is. Unfortunately, I've experienced this with quite a few places.

# The overlap between data science and data engineering

9:41

Alexey

**I guess, as a data scientist, you already needed to do some things that data engineers would do, right? You mentioned that at SoundCloud, you not only did modeling, but also everything that was before modeling, like building data pipelines, and after modeling – deploying the model. How does what you did as a data scientist overlap with what data engineers usually do?**

10:07

Ellen

Yes, there's a lot of overlap. As a data scientist, you often don't work with perfectly clean and perfectly delivered data. You will still build up your own pipelines to make the data accessible, especially as you move into more production-level things. I mean, data scientist is a very loose title and what you do in this role can be very different. There's often a lot of pipelining work that you do yourself. I think it's a good thing that data scientists do that and don't just rely on data engineers to handhold them through these steps. There are a lot of transferable skills. Also, data engineering is a very broad topic and I never did more Kafka real-time data engineering kind of stuff. I've carefully navigated myself around that. I've always done what is now considered analytics engineering, which is more like preparing data for BI and data warehouses and scheduling, batch processing, and all these kinds of things. That was more my realm of data engineering.

There's this whole other universe of data engineering that I've never really touched, where you need a better understanding of distributed systems and things like that, which I don't have. That is often more the playground for people with a really strong distributed computer science background. That just is not my realm. If anything, that's more in the realm of what is now called analytics engineering – that kind of branch of data engineering, I think data scientists are really well prepared for. There are obviously things we need to learn, but I've often found that people that come from pure software engineering don't have a feeling for data – that they struggle more to move into that space than the people that have a data science background. They just need to level up on software engineering and collaboration skills.

12:02

Alexey

**What is this “feeling of data” in your opinion? What is that?**

12:06

Ellen

Yeah, that means understanding, it's a lot of things about data. Data, as you know – I don't have to tell you – it’s very complex. How it's produced, all the quirks to test, the statistical aspects of it, understanding what the data actually means, how it's structured, how it evolves. From software engineering, I remember that data is often just something that we don't really worry about – it’s just something we pipe in and out, but we don't really worry about what it is.

12:33

Alexey

**Some JSON or XML file – you just get it, do something with it, and then you spit it out. And then something happens after that. Right?**

12:42

Ellen

But the big part of data engineering is really worrying about the quality of your data, that's probably the biggest challenge for data science. That's something that data scientists are very familiar with – data quality issues, how to deal with them, what to do about them, and having these conversations with the people that collect the data, produce the data and all these kinds of things.

13:05

Alexey

**You also mentioned that you were doing more work of what today is called “analytics engineer” rather than a Kafka distributed system kind of data engineer. Do you think this worrying about data quality applies to all data engineers, regardless of whether they work with distributed systems, or more with DBT kinds of tools?**

13:28

Ellen

I think it applies to everyone because data quality is just a theme of data. I mean, data always has quality issues. You can’t avoid that. I think just the extent that people worry about it is different. Maybe it's because I'm not that in tune with that community, but I hear fewer concerns about data quality from the distributed systems crowd than I hear from the analytics engineering crowd, but data quality issues are just a fact of life that you have to deal with.

13:55

Alexey

**To summarize: the skills that are transferable from data science to data engineering are 1) This pipeline building thing – you need to prepare your model or the data that before you can put it in the model, so you need to have a data pipeline. 2) This “feeling of data”– knowing that data is not just a simple JSON. Is there something more?**

14:23

Ellen

Yeah. Generally, the whole explorative and communicative approach – data scientists usually have to work really closely with their stakeholders and their business demands. That is super important as a data engineer as well. You never produce the data for yourself and you never produce it yourself – you always get it from somebody and you get it for somebody. Talking with both the producers and the data consumers is really key, as is being able to understand what they need, even if they don't know exactly what they need or why they're producing the data and how they're producing the data.

# Skills to learn and improve for data engineering

15:02

Alexey

**Okay. So: pipeline building, getting a feel for the data, and stakeholder management and communication. But what were the things where you needed to upskill yourself or to learn in order to make the transition? One thing you mentioned is that as a data scientist – and for most data scientists – it's not the focus to produce good quality code. I guess that was one of the areas that you needed to upskill yourself, right? Are there other areas?**

15:41

Ellen

There were two other areas that I think are really important. One is the whole idea of collaborating with other engineers. There's a whole amount of tooling like version control, books, CD/CI pipelines, and all these kinds of tools that software engineers usually use to collaborate together and to make sure that whenever they work in larger teams that the code still runs, still tests properly, doesn't break and can be restored if one of them makes a mistake, and that they catch mistakes and all that kind of things. Usually, as a data scientist, it's very common for us to work alone on our own Jupyter notes or whatever we might be using. But there’s this whole collaboration around making sure that you can work with your team members in a reliable and consistent way. That's something I had to learn.

There’s also the mentality around pairing and code reviews and things that data scientists sometimes talk about and sometimes are aspirational about, but aren't that much part of their daily practice. The whole “needing clean code and good quality code” is just an aspect of that, because you don't write it as an end in and of itself – just to make the code really beautiful so you can put it in a frame on the wall. It’s really to help your colleagues understand what you were thinking about when you wrote that code. The other thing is, the whole “how to deploy things,” and the whole DevOps aspect of data engineering, which is pretty strong actually, how to how to deploy things, how to how to spin up and shut down servers or Airflow clusters, or whatever you might need to put it another spot, to dealing with our cloud infrastructure in an efficient way, and not just by poking around in the UI, but also by automating things efficiently. That is the other thing that I really had to learn. That was my least favorite part initially, but now I enjoy it.

17:34

Alexey

**Yeah. This poking the UI part is very close to my heart as a data scientist, because what we are trying to do at OLX is to educate data scientists to use things like Terraform. It's always easier just to go there, click buttons in the AWS console, and then you have your lambda function or whatever. But then people who need to look after the infrastructure – after the AWS account – they always come and say, “Hey, what are you doing here? Why did you do this?” Because they don't have any visibility into what's going on.**

**Do you think that for data scientists – people who do not plan to switch to data engineering in the future – do you think it's also useful to learn skills like CI/CD testing, infrastructures, code automation, and all these things? Do you think it's useful for them or that it shouldn't be their focus?**

18:31

Ellen

I think the data scientist role is a bit bifurcating right now. There's more ML engineering and MLOps and all these new fancy titles that are springing up. For everybody who wants to work more on this building production service ML, it is definitely really important to understand how to do monitoring, how to do infrastructure, automation, testable infrastructure and all these kinds of things are really, really, really important if you want to have anything in production. Because if it's in production, it has to meet the quality standards of everything, otherwise if it's in production, it can also be the weak link in your ecommerce or whatever infrastructure you have. If you really want to stay in a more research focused thing, where all you do is prototyping, which I guess still exists, I'm not sure. It's becoming less and less common to have these kinds of data science roles where they’re really just putting prototypes of visualizations. There I guess you don't need to have these skills, but I think the trend is moving in the direction where it's really valuable to have the skills.

# Ways to pick up and improve skills (advice for making the transition)

19:36

Alexey

**This part of modeling is only a small part, right? But the vast thing before that is “How do you prepare data?” Then a vast thing after you train the model is “How do you go about deploying this?” The modeling part is only like a couple of percent of the actual work. That's why we have ML engineers and data engineers – to help data scientists take care of that. Yeah, interesting. How did you actually pick up the skills? Did you just learn by doing projects or did you learn it at work? Or did you need to take some courses? How did you learn this CI/CD and all these other things?**

20:23

Ellen

I mostly learned it at work. Native Instruments and at ThoughtWorks I was lucky to have worked with really talented people and ThoughtWorks, in particular, has a really strong culture on engineering practices, so I picked up a lot there. But also at Native Instruments, I had a colleague who was really dedicated to these kinds of practices and brought a lot of this into our team. I was fortunate to learn this, but back then, when I moved into data engineering, there weren’t that many courses around these topics. Therefore, it was kind of a necessity to learn it at work. But I think nowadays, if you have the chance to take one of those courses, you should. Right now, our data engineer in my team is currently taking the data engineering bootcamp that you're organizing. I think opportunities like that are really helpful for people to pick up these skills in a more organized fashion and not just rely on the lucky or unlucky coincidence that your colleagues can help you.

21:25

Alexey

**But do you know how it usually happens? Yes, indeed, we don't have a lot of materials for data engineering. For data science, we've had courses for quite a while – for five or more years. For data engineering, this is still emerging. Do you think if a data scientist wants to become a data engineer, that they already have enough skills to get the job as a data engineer? Or do they need to upskill themselves before they can get a job?**

21:55

Ellen

It's very individual-based. I've seen data scientists that are brilliant and really pick up software engineering and DevOps really quickly. And I've seen people that really struggled with that. I think that's why it really depends on the individual. I would always recommend trying it first. Also, before you make a big career transition, figure out if you really want to go in that direction. I think it's always helpful to do a side project or try it out at work in a small context where you can find that out before you decide that you really want to change your career. That's a mistake I've made, actually, sometimes to switch roles too quickly and then realize it wasn't for me.

22:33

Alexey

**Is that what happened with data science?**

22:35

Ellen

Yeah, exactly. For me that was a bit of a dead end. I probably could have figured that out earlier, but I didn't. I would recommend for people to try it out and see if they would consider a new role more attractive. That way, they can really jump into it without switching their entire job.

22:54

Alexey

**Probably for data scientists – at least for most data scientists – they have a way to actually try this. You need to take care of data preparation, right? Even if you have data engineers in your team this still needs to happen. Maybe you can just work closer with data engineers and learn from them. Then you can realize if this is indeed for you and only then decide to transition to a data engineer full time.**

23:25

Ellen

Exactly. Maybe then also invest more fully into formalized training or courses or something like that. I think that's the best path – to just try it out by expanding your scope of work a little bit, seeing if it's for you and how comfortable you are and how much joy it brings you.

23:41

Alexey

**I see two major paths for people who want to get into data science. Usually, they either come from more mathematical backgrounds – maybe they have a PhD in physics – in other words, they come from academia. Then there is another path, which is people who are software engineers who want to get into data science. I guess there is also a third way now, which is people who graduate from universities and become data scientists immediately. I think this is also a thing right now – for many people data science is their first job.**

**For those who are software engineers and become data scientists, it's not that difficult to then transition to data engineering because they already have all the necessary skills. They know how to use the terminal properly, they know batch, they know CI/CD – all these things that are needed for data engineers. But what about the people who are coming from academia or those who have data science as the first job – how can they actually level up their software engineering skills? Do you know if there is a good course about that somewhere?**

25:04

Ellen

Yeah, there are good courses about this. Data camp is not really recommended anymore for various reasons, but that used to be my go-to place because they had a really good engineering track. But there are other online courses that focus on these things. I would always recommend for data scientists who want to level up on programming skills to take one of those intro to software engineering courses, and even if it's web development or something totally unrelated – something that they may not really be that interested in. But something that's not geared at data scientists because in the programming for data scientists course, they usually don't put a lot of software engineering fundamentals.

So they need something that's more of a track to become a web developer or even an Android developer or something like that. They usually teach the software engineering fundamentals in those courses. So I would recommend trying that. It's always useful if you can go build a small web app or you can build a small Android app – it's not a waste of skill. That's maybe a better way to find out if you're interested in these kinds of things and then later purely focus on your Python skills and learning yet another deep learning library.

# What makes a data engineering course “good”

26:20

Alexey

**So what kind of things do you think a good course should contain? I guess, build tools – how exactly you build your software, right? Then testing, CI/CD, command line basics – how you navigate something like Linux, or how to use Linux. Bash. Is there something else that you would say is fundamental for all software engineers?**

26:49

Ellen

Two things, yeah. Git – which you probably included somewhere in build tools, but it's worth pointing out separately. Docker is also really, really useful. That may be a separate cause in many cases, but it's definitely a really valuable skill. Then there's the whole idea of just collaborative coding or clean coding – knowing the best practices on how you structure coded functions and objects or modules – these kinds of things. How you comment, how much you comment – all these things that software engineers pull their hair out and get into holy wars about. But it’s worth understanding. How many lines of code should be in a function? All these kinds of questions are really useful to get a sense for, even if you're never engaged in any of these holy wars.

27:39

Alexey

**So how many lines of code should there be in a function? [laughs]**

27:42

Ellen

Less than a screen [laughs] that’s my answer.

27:45

Alexey

**[laughs] Less than a screen. Okay. I remember reading the Clean Code Book from Robert Martin. I think his recommendation was like eight lines or something like that. This is pretty drastic, right?**

27:56

Ellen

Yeah. Exactly. Just enough to have one full if then/else than statement. That was his recommendation.

28:02

Alexey

**So basically, you have a function declaration, then just a few lines. Then you have “return” and that’s it. [laughs]**

28:09

Ellen

Yeah, he proposes having nesting of your function so that every line is pretty much a function call until you have your really small function. Really strong decomputization, which has its benefits and drawbacks, but yeah…

28:22

Alexey

**Do you think it's a good book nowadays? It's pretty old, right? It's more than ten years old I think.**

28:28

Ellen

It’s pretty old and it's very Java-centric. Unfortunately, I have not found a better book. I still keep thinking somebody should write a better book, or rather a new, more modern book about it. Maybe also something that’s less controversial nowadays due to politics. But unfortunately, there's still nothing better at the moment. I still really recommend it, even though I have a lot of pains about it. It's still, unfortunately, the best thing we have as far as I'm aware.

# Languages to know for data engineering

28:54

Alexey

**Even though it's Java, right? I started as a Java developer, and for me, it was eye opening. I also recommend this book to people, but I realize that now it may be outdated. Maybe that's another question – what kind of languages do we need for data engineering? Is Python enough? Or do we need to go with Java and Scala and perhaps other languages?**

29:22

Ellen

Again, it really depends on what kind of branch of data engineering you want to go into. If you go into something that’s more analytics, then yeah, I would recommend for data scientists to at least try it out first. SQL and Python are often enough, but sometimes you may need Java, depending on what scheduler you’re using. But usually you can get very far with Python and SQL and maybe JavaScript. But JavaScript is the lower ranking third option. If you want to go into the whole Kafka and real-time streaming and distributed systems branch of data engineering then yeah, Scala and Java, I guess, are unavoidable. I'm not saying they're terrible, even though I don't really like both of these languages, but it's just my personal preference. They’re actually really good languages if you get into them. I've done a lot of Java development and a fair amount of Scala development in my life, but I'm kind of glad I don't do either right now.

30:24

Alexey

**[laughs] It's the same for me. Well, at least for Scala. I don't know, there are too many ways of doing the same thing and not all of them are obvious, I'd say. To me at least. Why JavaScript though? Why do you recommend learning JavaScript?**

30:45

Ellen

It's becoming more than just a backend language as well. JavaScript is kind of emerging as a general purpose language in a way that Python always aspires to be but never quite made it because it doesn't have a front-end component really to the same degree. There is tooling also in the no code space, for instance, which relies on JavaScript as a scripting language. So it's useful to know, but as a really low ranking, I wouldn't give it a high priority that the other languages have. But it's useful to know – just to be able to read a lambda function that was written in JavaScript that your colleagues wrote and be able to see where you have to maybe modify it or at least talk to them where they need to modify it.

31:34

Alexey

**I guess the reason that these obscure JavaScript functions exist is because when you Google something – when you try to look something up – often the examples are in Python or JavaScript. When I see an example in JavaScript and I need to do something quickly, I just copy/paste, check that it works and then I forget about it until it stops working.**

31:55

Ellen

Exactly. I think even in BigQuery, for instance, the default UDF language (user defined functions) are just JavaScript.

32:05

Alexey

**There is no Python support?**

32:07

Ellen

I haven’t seen it, no. I think we had to write all our UDFs in JavaScript for some reason. JavaScript pops up at the weirdest places. That's why it's useful to know.

32:20

Alexey

**Probably, Google wanted to have broader coverage. I don't know what the latest status is, but I think if you take software engineers in general – people who know how to code – there are probably more people who know JavaScript and Python, right?**

32:40

Ellen

Definitely, yeah.

# The easiest part of transitioning into data engineering

32:43

Alexey

**Perhaps that just gives some wider coverage to Google – to BigQuery. Okay. What do you think was the easiest part of your transition?**

32:53

Ellen

The easiest part of the transition was how much demand there was for data engineering. I think that still holds. Even if you're not the greatest data engineer when you're starting out, people will still get really excited about the fact that you exist and that you applied to their company. It's very easy to find a job in data engineering. It's not super competitive, which is a good thing when you're starting out. It also has its drawbacks because you can easily find yourself in a situation where you're way overwhelmed and expected to do things that you aren't ready for and where you don't have good mentoring in place that can help you in your job. There may not be enough senior or experienced people around that can help you with what you're doing. But the easiest part is definitely – just give yourself the title of data engineer and your LinkedIn profile invites will explode.

33:45

Alexey

**Interesting. I observed a similar thing, not with data engineers, though – with infrastructure engineers. We call them “site reliability engineers” but another name is DevOps engineers – people who take care of infrastructure. It's super difficult to find these people. When we open a position for data science, in one day, we have 100-200-300 applications. For a week, it can be quite a lot. Our recruiters just closed the position after a couple of days. But when it comes to site reliability engineers, we open a position and nobody applies. The second day, maybe one person will apply. The third day, again, nobody. So HR needs to actually reach out to people on LinkedIn and ask them “Hey, consider our position.” I guess to some extent, this also applies to data engineers.**

**I think data science still has this marketing thing like, “It’s the sexiest job of the 21st century.” So people get excited and everyone wants to do that. But I think data engineering is also getting some traction. Now people like you realize that it's not what they want to do and people see the demand for data engineers. There are also quite a few posts on the internet, such as on Hacker News and on Reddit, where the headlines were “you don't need data scientists, you need data engineers.” You probably saw them as well, right?**

35:20

Ellen

No, I actually did not.

# The hardest part of transitioning into data engineering

35:23

Alexey

**This is the type of content that people on Hacker News usually like - very controversial stuff. Yeah. So that's interesting. What was the most difficult part of the transition for you?**

35:40

Ellen

The most difficult part for me was losing some of the autonomy I had as a data scientist, and then also having to work in these really close, tight-knit software engineering teams. I was really used to having a lot of space, working the way I wanted to, and not worrying too much about other people being in my space, other than my stakeholders – I knew how to work with them. But I was not familiar with working with other software engineers that much – at least I hadn’t done it in a long time. So when I got hired as a senior data engineer, things were expected of me that I had never done before, in terms of collaboration and in terms of leadership and leveling up on those was quite hard. That's not something that you just pick up by reading a blog post. But that was really a very different way of working and not even understanding what was expected of me, because the people I was used to working with were data scientists, so they didn't know what to expect of me either. Then there was this whole mismatch of expectations on both sides. I figured it out, but it was hard.

36:43

Alexey

**Can you maybe give an example? Is it like setting up the way you do things, picking up some frameworks, or what exactly do you mean by this leadership?**

36:57

Ellen

Yeah, it wasn't technical skills at all. I could pick up the tooling that worked with most of it before and I could pick up on what was missing easily. It was more like, “How do I communicate? When do I communicate? How do I pair with people?” Pairing was probably the hardest part for me to learn because I'd never done that before.

37:16

Alexey

**Paired programming, right?**

37:17

Ellen

Yeah. Paired programming, yes. Work was really heavy on emphasizing pair programming, but sharing mainly just boiled down to sharing my thoughts and knowing when to ask for help and when to offer help – and all these kinds of small things that I wasn't used to, since I was working mostly by myself before. Really, it was knowing how to work very closely with people on a day-to-day level, keeping them in the loop, knowing when I need to get myself into the loop, and all these kinds of questions.

37:49

Alexey

**So you would say that data engineering is more of a team sport than data science? And engineering in general, I guess. Because usually, you have a lot more engineers than data scientists, for a team, you maybe have one data scientist, and then a bunch of backend engineers, right?**

38:08

Ellen

Exactly, exactly. And usually you also have more data engineers than data scientists. So there's a whole different approach to working together.

# Common data engineering team distributions

38:20

Alexey

**In your experience, do data engineers usually work in one team? For example, maybe there is one platform or data engineering team? Or are they spread across different teams?**

38:33

Ellen

I’ve seen both options. I prefer the model where there's not a data engineering platform team, I think. Unless you have a really large company where you really need a data platform infrastructure and you need people dedicated to that thing. In smaller companies, I think it works better if the data engineers are embedded either with the other data folks or as part of a wider platform team. But what I've usually worked with is kind of embedded into the more general data teams that consisted of analysts and data engineers and data scientists – maybe ML engineers, and all sorts of data specialists.

39:11

Alexey

**In this setup you usually see more engineers than data scientists, right?**

39:18

Ellen

Yes, usually. But there can also be a lot of analysts, for example. That's a common thing to see. It depends on the setup.

# People who are both data scientists and data engineers

39:30

Alexey

**Okay. There is a question. Do you know if there is a name for the role for people who are both data engineers and data scientists? Does such a thing even exist at all?**

39:44

Ellen

Um, I've not encountered it as such, but I think the closest I've seen is really an analytics engineer. Again, it depends on what you consider a data scientist as being a data engineer because the overlap might be on all sorts of things. If it goes to the end, it could also be an ML Ops engineer, for example. That could also be an intersection between a data engineer and a data scientist. There are a bunch of titles, but it really depends. Maybe the person could clarify a bit. It really depends on what you're doing at this intersection.

40:22

Alexey

**I think, as a data scientist, you – or at least I did – need to do a lot of data engineering. The data is not just magically a CSV file that you can use that is clean. You need to do a lot of work before you can put this into a machine learning model and train your model. For me, I even needed to set up a workflow scheduler and to do all that.**

**I think in startups, it's pretty common that they hire a data scientist and then it turns out that this data scientist actually needs to do data engineering work before they can start with the data science. I also saw a title in LinkedIn – some people put this title – the title is “data science engineer.” I don't know how common it is or if it’s even a thing, or maybe they just decided to put it there because this is how they felt. I don't think it's a common thing.**

41:25

Ellen

I've seen that too. I haven't seen it in a job description. I’ve only seen it in people's profiles.

# Pet projects and other ways to pick up development skills

41:29

Alexey

**Yeah. It’s probably people who ended up doing data stuff even though they were hired as data scientists. Okay. Chetna is asking if you have any tips for people who do not have development experience – how can they transition to data engineering? I think we talked about that already. It involves picking up all these skills, like general engineering skills. I think it was build tools, testing, CI/CD, Git, Docker, clean coding, command line, testing. Is there anything else?**

42:14

Ellen

Yeah. Especially if you don't have some engineering experience, I would recommend just doing pet projects on the side. That's very common for software engineers to do. They just build something random that they think is fun or useful. Find some friends to work together with, because that makes a big difference if you're not just working by yourself, but if you're also working with two, three other people. Pick something that you really find fun. For instance, I’ve built a lot of Twitter automation things back when I was trying to get into data engineering. I didn't use any of the skills I learned about Twitter and Twitter analysis at the time. But it was really useful to learn to deal with annoying things like ORs and figuring out how that works and using Git properly and all these kinds of things. If you have time – not everybody has the luxury to have that kind of time – but if you do, it's really helpful.

43:12

Alexey

**What was one of these tools? Maybe you can give us an example? Is it like pulling data from Twitter for doing some analytics, or something else?**

43:20

Ellen

Yeah. That was a while ago. That was before Cambridge Analytica, but I read the papers that Cambridge Analytica was based on – Kosinski papers from Cambridge University. He did a lot of identifying Big 5 profiles out of Twitter data and since I was a psychology student at the time, I wanted to see if I could replicate it. So I pulled data from Twitter and that was supposed to be turned into a visualization of the Big 5 features as a web app. We never finished that thing, but that's kind of the direction we wanted to be going.

44:00

Alexey

**Okay, yeah. Thanks. A question from Harry, “I am currently in the same station. I am a data scientist and I want to move into data engineering. Would you recommend doing projects that depict real-life data engineering tasks? Do you think it would help in getting jobs?” I think the short answer is “Yes,” but maybe you can also give a longer answer, like what kind of projects can be done in addition to what we just discussed – like pulling data from Twitter. What else can data scientists do in order to move into data engineering? What kind of projects can they do?**

44:41

Ellen

Generally, if you do side projects, I think it's really helpful to not think about what is the most marketable but really, what is the most fun to you, because you need to keep up your motivation for a while. Usually that comes if you build something that's interesting to you, even if it may not be the most marketable thing. For instance, there’s a project I've seen a friend build when the pandemic was fairly fresh – he had a computational biology background and he wanted to help with identifying genetic markers for vaccines. As you can see, that was a while ago. He got some datasets around the COVID genome and some other genomes and then he built a whole ML pipeline – data engineering pipeline – around extracting that data and building it up with CD/CI tools, and understanding data, translating it into different formats, and extracting the genome data, and re-encoding it. I didn't understand the biology part of it, so I can only give a very bad representation, but that was kind of what he did. I thought it was pretty cool.

45:47

Alexey

**One recommendation I usually give when people ask me, “How can I learn more about building data pipelines if I am a data scientist?” I usually suggest building a scraper. Let's say we want to build a model for predicting the price of an apartment or a car, right? We have a lot of websites that sell cars or where you can find apartments. You can set a scraper that goes to these websites every day, pulls the data from there, puts them in a CSV file, and then puts them in the cloud. Then you can schedule it with Airflow. So you can have multiple steps there and the first step is the actual scraper that would go there and pull the data. Then the other step is – you have the pages, now you need to process these pages somehow. Then you extract, or parse, this data. Then the other step would be to maybe put this in a CSV file. Then you have a CSV file in your S3 or Google Cloud, or whatever cloud you use. I think it's important here to use a cloud and to use tools like Airflow or other schedulers. And then one of the steps there could be taking all this data and training your model. Then it's not just a CSV file that you download from Kaggle and train your model. It's still a toy pipeline, but you have a pipeline that you schedule – that you run – every day, and one that you can use to actually learn all these things – learn Airflow or whatever other scheduler. Do you know of other similar projects? Does something come to mind?**

47:41

Ellen

Not really, no. But again, I think the general approach that you used was really good – about building a real life pipeline with following whatever best practices are. But really, I would reiterate the point that you should pick some data that you find interesting. That was probably the best advice I've been given by experienced data scientists when I got interested in this space. I asked about recommendations for projects and I was recommended to pick a dataset that I really wanted to find something out about. I think the same applies – if you build a data engineering pile, build it for something where you actually care about the outcome. You won't always have that luxury at work. Sometimes you build a data pipeline for data you're really not interested in. So at least when you're doing it in your spare time – care about the data.

48:29

Alexey

**I saw a post on LinkedIn about somebody who was looking for a flat in Berlin, which is not very easy these days. What they did is also built a scraper. They would look, “Okay, where are the flats?” Basically, they get all the data first, and then they see the flats with the price they are interested in, the areas, the neighborhoods they're interested in. Then they look at flats that stay there for a while and this way, they use this need to find a flat. Based on that, they built this scraping pipeline and it helped them to find a flat.**

49:19

Ellen

That's awesome. That’s a really cool project.

# Dealing with cloud processing costs (alerts, billing reports, trial periods)

49:22

Alexey

**I don't know, maybe it's a bit off topic, but there’s a question from Brahm. “Most cloud platforms, data processing cost structures are not really transparent. Do you have any suggestions to manage data processing costs?” I guess this is also important when you learn to use these things. So do you have any suggestions?**

49:44

Ellen

I'm not sure that I would agree that they're not transparent. Usually the cloud providers provide very, very detailed billing information if you click into the billing console. Especially if you just have a side project, you can easily burn a lot of money if you don't shut off your instances and they keep running or whatever you might be doing.

My recommendation is – most cloud platforms have a budgeting alert for each function, so if you want to manage your costs, find out where you can get you an email and get a warning for whatever – half of the money that you're willing to spend, or a quarter of the money that you’re willing spend. That way you get pinged via email. Other than that, really dig into the billing console, because there's a wealth of information in there. You can run it down to the nitty gritty details of what exactly you're spending on – each processing cycle of your AFL cluster. That information is there. It's a bit of a science in and of itself to dig into that information. It may be a good data analysis project, just to figure out how to extract what you're interested in from your billing reports.

50:59

Alexey

**You can pull this data, right? Then you have another pipeline to practice.**

51:05

Ellen

Exactly. You can build a pipeline – it's very meta. You can build a pipeline about your pipeline costs. [laughs]

51:12

Alexey

**[laughs] I don't think this is a thing in AWS, but in Google Cloud, you have a trial period where they don’t charge you any money for the first couple of months. They give you some money – some free credits – and you can do whatever you want. If you run out of these free credits, they don’t start charging you. They send you an email saying, “Hey, you’ve run out of credits. Do you want to upgrade or not?” It's a pretty safe environment to learn things there. In AWS, I don't think that's the case. They do have a free tier, but some things are part of the free tier and some things are not. If something is not a part of the free tier, then you need to spend money on it.**

**You also need to be careful – let's say if you spin up a Kubernetes cluster or something like that, you need to be careful once you've done whatever you wanted to turn it off. Or else, you'll get a bill at the end of the month that you won’t like. I think the support is pretty good. Some of the students had these problems – they forgot to shut down a Sage Maker instance with a GPU, which is quite expensive. Then they just wrote support saying, “Hey, I accidentally forgot to do this. Would you be so kind as to remove that?” And they actually did, saying “Okay, yeah. Things happen.”**

# Advice for getting into entry level positions

52:46

Alexey

**Other questions. “In your perspective, what amount of project experience should we get to start applying for entry level roles in the industry?”**

52:57

Ellen

That's a very generic question. Would it help? Is this for data engineering?

53:03

Alexey

**Let's imagine that a student has a general software engineering background. So they probably learned data structures, algorithms, programming, so they also know SQL. Now they want to start working as data engineers, what do they need to do?**

53:31

Ellen

Especially if you have a relevant background in studies, I would always recommend going directly for the entry level positions because I'm not a fan of people who have a full degree and then start to get into unpaid internships or doing a lot of side projects just to get into the field. That may happen to career changers, unfortunately, quite a bit. But if you have a relevant degree, then definitely try to aim for the entry level positions directly.

You will have done enough coursework and enough projects in university to be employable. Not every company offers entry level positions in data engineering, but those that do, they will be prepared for graduates. Or if you graduate from boot camp or something like that, then I would also directly aim for an entry-level position. If you don't have a relevant degree, but you have some other degree, then try to not do so many projects in your spare time, but try to get internships. That work will be a lot more relevant in terms of experience you get quickly, rather than just trying to motivate yourself. I've seen people that have purely graduated from their own projects, but those people are unusually driven. I've seen a lot more people fail with that approach than I've seen people succeed.

54:58

Alexey

**You mentioned that there are not so many entry level positions for data engineering and I think it's true. At least this is my perception, usually data engineering companies want to see experienced people – as you mentioned, especially if you're talking about Kafka and things like that. They already need to have solid software engineering experience and know a bit of distributed systems before they can be hired to this position. Do you think this is the case? Let's say that since there are not so many entry level positions, would it be a good idea for people who graduate from university to first work as a backend engineer before they start working as a data engineer?**

55:46

Ellen

That's a really good question. I think it used to be that pretty much every position that I've seen for that was at least mid-level, if not senior, because the companies started out with building the data engineering teams. I think that's changing now. There are increasingly more entry level positions, but it depends on the type of company. Unless you're really confident in your skills, or maybe you want to find your own thing – that's an exception – but otherwise, I would really recommend starting in either a consultancy or a larger company that already has an established data engineering department.

For instance, I worked a lot in my life in agencies and consultancies, and those were always good career accelerators for me. I've seen the same happen for people that work in big tech companies like Zalando, Delivery Hero – all these big building companies, or whatever the local equivalent might be. Those tend to be career accelerators because you usually have a really well-structured learning environment. They have enough seniors and they're prepared to take on juniors. They are prepared to mentor and develop their juniors. It's very hard if you're starting out – and this is something that I’ve also seen quite frequently – if you're a junior and you get hired as the first junior somewhere. There may not be a senior around and they just expect you to start. It sounds really exciting, but in most cases I've seen that happen, it was a recipe for frustration on all sides and a stalled career.

57:22

Alexey

**Interesting. So consultancies are good career accelerators. Perhaps it’s because – let's say you have clients, and I imagine that there are not enough seniors to work with all these clients, so that's why they have this training in place to take this workload from seniors and put them on juniors. That's why they want the juniors to be ready to do the work as fast as possible, right? Then you have many projects – you need to be able to move quickly from one project to another and that's how you get to see a lot of different things. Is that right?**

57:59

Ellen

That and it's also a very financial thing. Juniors simply cost less. With a consultancy, there's a big financial incentive for them to have this pyramid structure where they have a few seniors that do the architectural work and mentor the juniors, but the bread-and-butter work is usually done by juniors. So, most consultancies out there have their business model around up-leveling juniors. That leads to a lot of them having structured entry level programs. There are a lot of expectations around mentoring and that's ingrained into the culture.

# Which cloud platform should data engineers learn?

58:36

Alexey

**Yeah, thanks. Maybe the last question. If somebody wants to learn cloud – should they go with AWS, Google Cloud Platform, Azure, or something else?**

58:48

Ellen

I don't think there's a real [difference]. I don't think it matters that much because they're all very similar, actually. If you know your way around one cloud, you can easily find your way around the next cloud, at least in my experience. Either find the one that's used at your company and get really comfortable with it and dig deep into it – learn all the functions that you can get a hold of – or find one that has the best free options that are relevant to you and just learn that one.

59:20

Alexey

**I guess another option would be in your city or in your country – look at what is the most common one, or the most popular one. For example, in Berlin, if I look at job descriptions, I think AWS is more popular than Google Cloud. Maybe if you're in Berlin, then going with AWS makes more sense. But I’ve heard that in other cities in Germany, maybe Azure is more popular than AWS.**

59:46

Ellen

It depends on the sector of the industry you're working with. A lot of startups use GCP because they have very generous startups offerings. For instance, we barely pay anything for our Google Cloud offerings. A lot of larger tech companies or more established tech companies that are not traditional enterprises, but are grown and mature startups – a lot of them use AWS. In Germany, a lot of the enterprisey companies use Azure. So it really depends on which branch of the industry you want to be in or which one you happen to be working with.

# Finding Ellen online

60:21

Alexey

**Okay, thanks. Before we finish, how can people find you if they have questions? LinkedIn, Twitter?**

60:30

Ellen

Yeah – either way works. Is there a way you can share my contacts with the audience?

60:35

Alexey

**Yes, I will. I will put this in the description.**

60:39

Ellen

Cool. Yeah. I can just share my Twitter and my LinkedIn profile with you and then I'm happy to talk if people want to get in touch.

60:47

Alexey

**Okay, then I guess that's it. So thanks a lot for joining us today. Thanks a lot for sharing your experience. Thanks, everyone, for joining us and watching this – for asking questions. Yeah, that would be it.**

60:59

Ellen

Thank you. That was really fun. I really enjoyed this conversation.