1:10

Alexey

**This week, we'll talk about quite a few things – building a machine learning platform, career, being a generalist or specialist – quite a wide range of topics. We have a special guest today, Krysztof (Chris). So Krysztof is a seasoned engineer with 17 years of professional experience in building software. Currently, he works at Zalando, where he's supporting machine learning practitioners as a machine learning platform engineer.**

**Actually, maybe you know that Chris gave a talk about that a few months ago, where he presented the machine learning platform they built at Zalando. We will include the link in the notes, so if you haven't seen it, check it out. It's in our channel too. Right now Chris is on a sabbatical, where he is learning and broadening his machine learning skills. Welcome. Pleasure to have you here again.**

2:09

Chris

Hello, everyone. Hi, Alex.

# Chris’s background

2:12

Alexey

**The questions for today’s interview were prepared by Johanna Bayer. Thanks, Johanna, for your help. Let's start. We'll start with your background. Can you tell us about your career journey so far?**

2:25

Chris

Yes. It started quite a while ago. In 2000, I built my first website and since then, in 2005, I started to work as a software engineer. I worked in pharma for a company called Roche. This was my first job as a software engineer, specializing in front end. Then I work for a company that some of you have fond memories of, called Nokia. I was working there on geolocation services, also working mostly front end and map applications and navigation.

Then I decided to switch my career to something completely different and got into game development. I was working for a company called Wooga. It's a Berlin-based startup, a wonderful place where I spent five years working on mobile games, for example, for iOS. I also built again, for this device [shows watch] for Apple Watch. It still runs on my device.

3:24

Alexey

**You can actually play games on the Apple Watch?**

3:27

Chris

Yes, that was one of the things we wanted to try out. And it's definitely possible. So the games are running even now,

3:33

Alexey

**Do people actually play it? I guess it's not really convenient to use it? Or is it?**

3:39

Chris

Part of the experiment was to find out if it is a good platform for games. It is possible, but there are only certain kinds of games you can build for it that make sense. For example, so-called “idle” games, where you do something and then you wait, and in a while your watch will remind you to play. It was more like a proof of concept.

3:59

Alexey

**So you cannot really play Tetris or Bubble Witch or anything like that?**

4:03

Chris

I think Tetris would be possible if it's a simplified version, but maybe not the most convenient.

4:08

Alexey

**Okay, interesting.**

4:11

Chris

And then after Wooga, I decided to switch careers again and joined Zalando. Zalando is a company where it's quite easy to switch between different roles and the company encourages that. I started as a full stack engineer working mostly with Python and then I moved towards data in the first team. In that area, I was working on data governance topics for one year. Then, three years ago, I reached my current destination, which is machine learning. I joined the machine learning platform team where I was working as a software engineer and now as the internal consultant.

4:49

Alexey

**You started your career, I guess, with HTML and a bit of JavaScript, right? [Chris agrees] You built your first website, I guess maybe with some PHP. Right?**

4:58

Chris

PHP and the very first language I started with was Perl, if you remember that one.

5:02

Alexey

**Perl. I know it existed – still exists – and every time I copy a snippet from the internet with some Perl code, my brain just explodes. I don't know how people in their right mind can write this. [chuckles] I hopefully did not offend any Perl lovers. I know they exist. It’s not for me, I guess. [chuckles]**

5:23

Chris

I actually really like Perl. It has a reputation of – some people call it “write-only language,” because you can write it but then it's not very easy to read.

5:30

Alexey

**“You live only once language.” [chuckles]**

5:31

Chris

Yes.

5:34

Alexey

**So then it was Pearl, and then you joined Nokia. What kind of stack did you use at Nokia? I guess Java or something like that?**

5:39

Chris

At Nokia, I was working on the front end applications that were running in a browser, both on mobile and on desktop. We were using JavaScript.

5:49

Alexey

**JavaScript, right. So you weren't writing any mobile code, only the front end.**

5:54

Chris

At the time, at Nokia, yes – it was only JavaScript. Actually, this was also the reason I was hired for this gaming company, Wooga, because we also tried to do mobile development using JavaScript. The first project was in JavaScript, and only later we decided it's better to use something native or Unity.

6:14

Alexey

**You were talking about Wooga, [Chris agrees] as a game developer you used Unity there.**

6:24

Chris

I didn't use anything myself. When I was there, it was a very exciting time because Wooga was trying different technology. We started with HTML5, which was a big thing at the time, around 2012. We built and shipped again, but we decided it's not really the most convenient and performant tool for mobile development, so we switched to native. I was mostly working there with Objective C and Swift. And then the company moved to Unity. But around that time, I also left and moved to Zalando.

6:58

Alexey

**And then at Zalando, as a full stack engineer, you said that you used Python.**

7:04

Chris

Mostly Python, yes.

# Switching careers multiple times

7:05

Alexey

**From what I hear, these are completely different stacks. So how did you change your career so many times? How did you convince the employers to actually give you a chance?**

7:23

Chris

Yes, this is an interesting question because sometimes when companies are hiring people, they look for experts and specialists in one area. But in practice, pretty much in every place I worked at, it turned out that in the end, the company is changing direction and they actually want people to learn new things. In every place, I was fortunate that the company allowed me to try something new and learn. At Wooga, the company decided to change direction, but they were happy with me as a general purpose software engineer and I was also very eager to get deeper into Apple’s stack. I got a few weeks to just get into the new codebase and then you just grow expertise as you're building software.

8:10

Alexey

**As I understood, you joined Wooga as a JavaScript developer – or front end developer [Chris agrees] – but then there was a chance to work on mobile stuff. And you took this chance, right?**

8:20

Chris

Yes. I was already hired with the intention to work on mobile development with JavaScript. The project was already going on. But then we decided that we actually don't want to continue working with JavaScript – we wanted to use native tech because it just performs better.

8:38

Alexey

**You were using Objective C and Swift and then somehow you ended up as a full stack engineer using Python. How did that happen?**

8:51

Chris

Exactly. Web was always very close to me and even though I mostly specialized in front end, even for my own private projects, I already did some back end. For example, I did PHP, Perl, which I mentioned, and also Ruby on Rails. I had the general idea, which was good enough to get me through the interviews. In the interviews at Zalando, I had to use JavaScript. But then the project also required Python knowledge. Then, at Zalando, I remember I forced myself to have a very quick learning experience with Python. I read some books about this, and started to code also in my spare time.

Very quickly, I was able to contribute. Also, what's worth adding is that a lot of experience as a generalist can transfer it from one tech stack to another. This was a very common theme in my career, for example, things such as SQL – one of the technologies that I used pretty much in every company, even though it was not always my expertise. Learning Git is also something that I just keep using over and over in every role.

# Freedom at companies

10:06

Alexey

**When you mentioned Zalando, I remember my interview with Zalando. Actually, I was interviewing for a data scientist position and I was asked algorithmic challenges, but I could use whatever language I wanted. My background was in Java development before data science, so for me, it was the easiest to use Java, and they were like, “Okay, whatever. Use what you want.” They didn't really test me for knowledge in a specific language. They just wanted to see how I could solve problems. I guess this is what also happened to you. In the interview they just check that you know some things and then maybe they didn't care much whether you know Python or very well or not. Right?**

10:54

Chris

That's right. Specifically at Zalando, we have something called “Tech Radar,” which is an idea that I think was initially invented by Thoughtworks, which is a consulting company. It's a map of different languages, libraries, and tech tools that we're using in the company. I don't know in what year you applied to Zalando. Do you remember when it was?

11:23

Alexey

**I don't know – seven, six years ago. But I had multiple interviews with them across years.**

11:29

Chris

We have changed it a bit since then. When I was applying, it was six years ago. Zalando gave much more freedom to teams to choose pretty much any language that the team wanted. Since then, we realized this leads to a bit of chaos when, for example, we have a team that uses Haskell and then another team is using Rust. It doesn't scale very well, although it gives you that freedom. Since then, we focus on preparing this Tech Radar, which has a list of languages that are encouraged in the company.

Right now the list is more limited. Java is definitely still there – Python, JavaScript, TypeScript, for mobile development, I think Swift is there. I'm not sure because I'm not doing that anymore. But the list is definitely more limited than before. You will still have freedom, and most likely you will try and work with different languages, but the list is not unlimited as it was before.

12:31

Alexey

**So if you're a Haskell enthusiast, maybe it's not the right place for you, right? [chuckles]**

12:36

Chris

I'm pretty sure we still have some legacy Haskell code, but it's not encouraged anymore.

12:43

Alexey

**Yeah, I wonder why. [chuckles]**

12:47

Chris

I got to know one person at Zalando (actually, he was even interviewing me) who is a huge Haskell fan. But it’s difficult to hire people who know Haskell. That’s one of the reasons.

# Chris’s role as an internal consultant

12:59

Alexey

**Okay. So you joined Zalando as a full stack engineer, then you worked in data governance, and then eventually switched to the ML platform as a software engineer, right? [Chris agrees] Can you tell us more about what you do in this ML platform team? Or you did, right? Now you said you’re an internal consultant. Maybe you can just tell us about your journey from that part in more detail.**

13:25

Chris

It's still the same team – ML platform team – but within that team, we have three smaller sub-teams or groups. I was first working as a software engineer, and the primary product of my team was something called the zflow. It's a Python library that we're using as a base building block for machine learning pipelines. I spent two years doing that. Last year, I switched roles a bit.

I still stayed within the ML platform team, but now as a consultant. I teach people how to use zflow. For example, I conduct trainings, and I also do a lot of user support and consulting in a way of understanding user problems and then helping them to architecture their pipelines and solve any issues they may have, either with our products, or with AWS tooling, which is another building block for us.

14:23

Alexey

**So how did you end up being a consultant? Is it something where you saw that data scientists do not know how to use zflow and then you saw the need to teach them? Or how did it happen?**

14:34

Chris

We had a consulting team from early on. The reason for that is that very often, with internal products, you build something but it's not very well supported. We wanted to avoid this mistake and that's why, from the very beginning, we put a lot of effort into things such as documentation, which is another thing that the consulting team helps with – training and onboarding, and then user support. We didn't want to just launch it and hope that people will use it and like it. We want these users to be happy with the product. That's why we have the consulting team. I must say that I like this a lot.

I like solving problems. And I must say, as an engineer working on zflow, I understood some parts of it very deeply, but I didn't have the big picture. I didn't know how people are using zflow and what the challenges they had were. It definitely also expanded my horizons and let me learn new things. Some people say that the best way to learn something new is to teach it. So that was definitely the case with it.

15:48

Alexey

**I also like saying that. [chuckles] That's true, I think. So what does your day look like as a consultant? What do you usually do?**

15:59

Chris

Yes. Most of it is user support. It's a kind of a responsive role. We have several chats where we answer questions. Sometimes the questions are quite easy to answer, but usually not. If someone comes to us, it means that they probably already checked our documentation and they have a more advanced problem, which requires going a bit deeper. Because we work mostly remotely, usually I just get a call with the member of a team who has this problem and we debug it together. That's one thing. Very often, if I see that it's a problem that happens over and over again, or maybe one that resulted from a lack of clarity in our documentation, I would just fix the documentation.

Sometimes, there is a bug discovered in the process. If it's something small, even as a consultant, I will try to fix it. But usually, I will just talk to the development team and they will fix it. We also have trainings. Some trainings are already happening and there is quite a lot of work involved with that because it's not only a passive presentation, but we try to do it in a very interactive way. People have to do their homework and they have to prepare, and then, during the training they have exercises and I will also try to follow up with them later if they struggle with the homework, for example – you take on an instance. That's the typical set of tasks I have to do.

17:37

Alexey

**I guess you need to prepare all the materials for the trainings, too? [Chris agrees] How much of what you do is hands-on? I mean, how often do you actually need to go and code?**

17:48

Chris

Not so much, I must say. I think this is a bit of a disadvantage that I don't code, especially when it comes to more complex problems – I can't solve them anymore as a consultant, compared to my previous role as a software engineer. I still submit pull requests, but usually they are documentation fixes. I also don't want to step over the competencies of my colleagues in the platform team who are actually building this. So there’s not so much hands-on coding.

# Chris’s sabbatical

18:26

Alexey

**Okay. And now you're on a sabbatical. [Chris agrees] Can you tell us more? What do you do on your sabbatical?**

18:32

Chris

This is a three-month sabbatical: January, February, March. I have to say, when I took it, my intention was to stay away from technology as much as possible because I've been working at Zalando for five and a half years, and as a software engineer, much longer than that. I thought I needed to break. The thing is most of my planned activities were non-technical, like focusing on my hobbies. But, especially last year and the current period is so exciting, in my opinion, in technology – especially in our field, in machine learning.

I have to say that I still catch myself and try to play with some things. Especially last year, three of the most exciting things for me were diffusion models. I am not a data scientist – this is not my background. I don't claim to understand the papers, even though I looked into them, and I even gave an internal presentation in the team about how they work. But I only have a high-level understanding of that. But especially as a front end engineer and someone interested in graphics, I find it extremely exciting.

Towards the end of the year, there was the release of ChatGPT, which is also extremely exciting. Even as a user, as a software engineer, or as someone who works in tech, I think this has a huge potential to change our jobs. We can also talk about that. So I definitely looked into that. Then there is something that is also partially related to what we do in the machine learning space and that is running code in the cloud. I was not paid to say this, but there is a small startup that I'm very excited to mention, which is called Modal Labs.

It's something founded by Erik Bernhardsson, who worked on the initial Spotify recommendation algorithm. He is also the original author of Luigi. Modal Labs is his latest project. You can think about it as a very easy way to run your local code in the cloud without worrying about infrastructure or about provisioning hardware. The idea is that you write your Python script, you execute one command to run it, and then it runs in the cloud. It's a bit like AWS Lambda, but much easier to use. They also made it very performant. So this is something that I started to play with this week and hope to dive a bit deeper during my sabbatical. That’s how I’m handling the non-technical. [chuckles]

21:16

Alexey

**[chuckles] So do you still have time for non-technical plans or that takes most of your time? [cross-talk]**

21:21

Chris

I have to say that I'm struggling. I’m trying to improve that. Three months sounds like a lot, but with my to-do list, it's really hard to fit everything in. It's a bit difficult.

21:33

Alexey

**Yeah, I also took two months off and I didn't notice how quickly these two months passed. It was just – I blinked my eyes and it's over. Go back to work. [chuckles]**

21:44

Chris

I drew the lines that I don't want to touch things that I do in my daily job at Zalando. If at some point I will find myself editing IAM policies, then it means that I crossed the line and it’s too deep into what I do.

# ChatGPT

22:01

Alexey

**You said that you're quite excited about ChatGPT as a developer and you wanted to talk more about that. When I saw ChatGPT, I was blown away. It was like, “Wow! So cool.” Now I use it for a lot of stuff, mostly around content generation – for posts on social media for, podcast summaries – for a lot of stuff in DataTalks.Club. I also talked to it sometimes.**

**For example, I read a book recently and I wanted to talk about the book with somebody, so I talked to the chatbot. It was pretty fun. I tend to use it very often. Recently, it was actually down for a day and it was like, “Okay, what do I do now?” I wonder – how do you use it? How do you include it in your workflow? What excites you about it?**

22:57

Chris

I'm still on the stage where I’m discovering what it can do. It's hard to find something that it cannot do. It can work like – the way I mention it now, or my metaphor for this is – if you know Data from Star Trek: Next Generation, or Jane, the assistant that the main character from Ender's Game had in his ear. They’re super-smart artificial intelligences that you can talk about anything with, and it will give you pretty good advice. I think we are still just discovering what it can do.

For example, it can do coding pretty well, so if you're stuck with a blank screen, it's excellent to just describe what you want and it will give you a decent answer. What was very surprising to me is that it can also work as a sparring partner for architecture discussions. I, and also some of my colleagues from the team, we gave it three advanced questions, for example, about Amazon SageMaker and IAM policies, and it gave very good answers. Here, there is a bit of a danger. Because on one hand, these are very good answers that sometimes are difficult to Google because maybe nobody else asked this very question exactly on StackOverflow, but it still requires that you bring some knowledge and experience to validate this answer. Sometimes it speaks with confidence, but it's actually wrong. My colleagues in the team and I were generally impressed by this, but we still know that you cannot trust it 100%. On the other hand, you can also say that if you talk to a fellow human, even if it's an experienced software engineer, they also make mistakes…

24:51

Alexey

**And they can also be overly confident, right?**

24:53

Chris

Definitely, they can be overly confident in fact. [chuckles] In fact, sometimes even more than ChatGPT. For example, one of the questions I asked was also answered on StackOverflow and it was a pretty highly voted reply, but I don't think it was a very good answer. The answer I got from ChatGPT was more humble, balanced, and rich in scope. So that's one of the things I use it for. But like you, I also tested this with just general content creation – things that are not technical.

You can use it as a friendly poet. It can write songs, not only in English, but also for example, in Polish or Russian. It can be a therapist. You can also talk to it about books. For example, it can give you a summary of pretty much any book that was written in the last decades, especially the classics, but also relatively newer books.

25:53

Alexey

**Interesting. Actually, I asked it to give a summary of a book and it said, “No, I don't want to do this.” It just said “I'm not good at doing summaries. But I can talk about the topic of this book.” I was like, “Let's talk about that.”**

26:08

Chris

So it could depend on the book. I tested this on one book and…

26:41

Alexey

**Maybe it depends on how popular the book is?**

26:17

Chris

Yeah. The book I tested this specifically was The Intelligent Investor, it's a book about stock market investing. It's pretty old. It's from, I believe, 1949. It's been around for a while, so maybe ChatGPT was more confident when talking about this book.

26:33

Alexey

**How do you actually use it during your sabbatical? I guess these are the examples, right? You discuss books and so on, right?**

26:46

Chris

Yes. I have to say that I still didn't fully incorporate this into my daily life. I think there will be more and more of that. Right now, I think the UX, especially for specific tasks, is not perfect. For example, if you want to write code, GitHub Pilot has slightly better UX, because it's just integrated with your ID. It's a bit more cumbersome to use ChatGPT. But I would say I'm still in the phase of discovery. For example, (we can add this to show notes) there is one person I highly recommend to follow on Twitter.

His name is Riley Goodside. He's possibly the first person in the world who was hired as a “prompt engineer,” which is a person whose main job is to write prompts to talk to AI. He's sharing a lot of his findings on Twitter and he's demonstrating, for example, how to trick these large language models into something absurd. But he also demonstrates what the possibilities are, and I think we're still in the phase where we're just learning what it can do. I think we'll discover much more this year.

28:05

Alexey

**Now, everyone’s getting on the hook from OpenAI and then once they make it paid then… we’re screwed. [chuckles]**

28:12

Chris

Personally, I look forward to the moment when they start charging for that, because currently it’s down pretty often.

28:19

Alexey

**I would pay too. [chuckles]**

28:20

Chris

I would pay, assuming that the price is reasonable. But I think it would be better for user experience. Maybe they will also have some free version for, let's say, students or open source.

28:33

Alexey

**They recently sent, or I don't know if it's them or somebody else – I saw a form where they were asking, “How much are you willing to pay? How much money do you consider expensive? How much do you consider cheap?” And I thought, “Hmm. No, I'm not telling you that.” [Chris agrees] Otherwise, it will be too expensive maybe.**

28:52

Chris

I thought about the same thing. I think I would be willing to be quite a lot, but let's see how much they decide to charge.

# How being a generalist helped Chris in his career

29:00

Alexey

**Going back to your background, you have quite a diverse background. You tried many, many things. Do you think it helped you with your career, for generating new ideas, for seeing new solutions? And how did it help you?**

29:16

Chris

I think, generally, yes. Like everything in life, there are pros and cons. The pro is that it allowed me to transfer different skills from my past jobs into new ones. I think sometimes it's just easier to start a new role, where I can just borrow from my past experience. I like to think about technologies that are good investments over the long term that I can transfer across different jobs. I mentioned a couple of them, for example, SQL is one of them. Another one is Git. Basically, these are things that follow the rule of “The longer something has been around, the more likely it is to stay for a long time.” I think there is a name for that. It's called the Lindy Effect. It's named after a restaurant from New York that existed for some time, called Lindy’s. And just the fact that it existed a long time means that it's more likely to stay around.

I tried to look at the things that existed for a while and it gives me more confidence that they will just keep working. JavaScript is another example. Now, JavaScript is almost 30 years old. This was definitely one of the good investments. Another thing that I'm actually learning or improving on right now in my spare time is getting better at Shell scripting and just using command line tools from a terminal. This is something that in every job that I work, the more I know about this, the more it pays off. So I'm trying to invest into things that I can transfer.

This definitely helped for being a generalist. I also think there are some general principles, for example – when I join a new project, I first try to understand what the user value is and how it helps the company to build something. So I try to have a product and user-centered view on the thing that we're building, instead of only focusing on technology. I think this generalist approach helps here. These are the pros. When it comes to the negative sides of being a journalist, I have to say that, despite spending many years as a software engineer, I constantly suffer from impostor syndrome. [cross-talk]

31:45

Alexey

**If you were a specialist, do you think you would not suffer from imposter syndrome? [laughs]**

31:52

Chris

Probably, people who are experts in something – deep experts – they also have it, because they also have major gaps in other areas.

32:00

Alexey

**You always compare yourself with somebody who knows it better than you. Right?**

32:06

Chris

Probably. Yes.

32:07

Alexey

**And it's not a good idea to do that.**

32:09

Chris

This week, I talked with two friends of mine. We used to work together on more than one project. One of them has been a software engineer for longer than I have and he says that he still suffers from impostor syndrome. I think that's just something that we have to embrace as part of our job that – regardless of how fast you're learning, technology is moving faster.

The gap between what I know as a software engineer or as a data engineer, versus what is there, it’s only going to grow, unfortunately. But one very practical consequence of that is that experts, or specialists, are able to build things faster. If you have a good match between a product that you need to build and expertise, then this person is definitely going to do it faster than someone who is like me, a generalist.

33:02

Alexey

**Interesting. In my experience, it's the opposite. For a project, you usually need a wide variety of skills. It's typically not just one person that you need. If it's an end-to-end project, which needs front end, backend, databases – all this stuff. Typically, if you're a specialist, and you know only the back end, it's probably more difficult for you to ask things. It depends on how niche the project is, right?**

33:34

Chris

Exactly. It depends on the scope. That's why I mentioned that if you have a good match between what is required for a project versus the skills that the team or the person brings to the project. The wider the scope, of course, the expert will struggle more when it's getting outside their area of expertise. But if you have something that is rapidly narrow in scope, then expertise will definitely help. I can give you an example.

Over a year ago, we had our internal API just going slower and slower – it was not really scaling well. Some requests were taking more than 10 seconds to return. I am not a database expert. I know my way around it, but I'm not an expert. But I was tasked with looking into that and improving that. I think I spent a couple of weeks on that. In the end, we got 10 times speed up in performance by optimizing our RAM layer for Postgres. But I'm pretty sure that a person who is very fluent in Postgres would probably do it in one day.

34:48

Alexey

**But with your background and your experience, you knew where to look for the answers. It just took you a bit more time to find these answers. At the end, you still did it, right?**

34:59

Chris

Well, I'm also quite sure that a Postgres expert would find even more than I did. [Alexey agrees] Unfortunately the team, we didn't have someone who was a Postgres expert.

# The cons of being a generalist and the importance of T-shaped expertise

35:14

Alexey

**Okay, so there’s two cons so far – impostor syndrome and experts can build things faster. Anything else?**

35:23

Chris

I was also thinking about how it affects recruiting and finding a job. Very often I saw that companies look for experts. Sometimes it could be a bit scary, like you see a job ad and you see, “We want five years of experience with large language models and you have to be an expert in this and that subject.” It's just impossible to meet all of that. But there is this common advice that you should often apply for jobs, even if you don't meet every criteria. I think this is good advice, because in every role and in every company where I was hired, I had to demonstrate some expertise, but in the end I was doing almost entirely different things. For example, I was hired at Zalando after a JavaScript interview, and then I actually learned Python on the job at Zalando.

I didn't have any expertise or any experience with data governance or machine learning, but this is what I do right now. It helps to have some expertise to get your foot in the door, but then you should be ready to learn. I think many people who are listening to us now have heard about the “T-shaped developer” or “T-shaped person” and for those who haven't, it means that like in the uppercase letter T, you have this one stem that goes deep. It's good to have one area of expertise, where you're really, really good at – an expert. But there is also this horizontal bar on the letter T. These are other skills that you should learn and be able to jump and go deeper into them, if needed.

37:08

Alexey

**What's the stem for you? The vertical part?**

37:14

Chris

The one that is closest to my heart is the web, front end, and everything related to HTTP and understanding how things work in the browser. I also tried to expand it. By now, it's maybe not a letter T, because I have a few of these stems.

37:32

Alexey

**Like maybe an E upside down?**

37:37

Chris

I’m not sure which letter it would be, Maybe an M. For example, I got pretty good at using Git. Very often, I help my colleagues to resolve problems with history, with rebasing, and things like that. I really also like Shell scripting and learning about different Unix tools. There's like an infinite number of them. The more you learn, the more productive you become. Then there are some things that are maybe less specific to one particular technology, like the ability to debug things. For example, the divide and conquer method for just narrowing down where the problem is.

Sometimes I'm used by my colleagues as a rubber duck – for rubber duck debugging. For those who haven't heard this term, sometimes it's very helpful to just explain the problem you have to someone or something, it could even be a rubber duck that you have on your desk. Explain the problem and after you explain it to someone, the solution just comes to your mind. So very often, I'm a buddy, or a rubber duck for my colleagues to help them debug problems. A lot of engineers prefer to build new things instead of debugging and finding bugs. I'm actually one of the few people who really like debugging.

39:03

Alexey

**That's a rare quality. [chuckles] Usually people don't like debugging. I don't really enjoy it, to be honest.**

39:12

Chris

I think most people don't like it. I treat it as a puzzle to solve.

# The importance of learning things you’re interested in

39:16

Alexey

**It’s like you're a detective and you're looking for a solution to a crime, right? [Chris agrees] That's pretty interesting. One thing I wanted to ask you is – what advice would you give to people who want to do something like you do, like change directions often? I think maybe I'll try to answer based on what you discussed so far and then perhaps you will add something. From what I understood from you, you said that it's a good idea to invest into fundamental skills – skills that are transferable, like SQL, Git, shell scripting, terminal, JavaScript, the ability to debug things – divide and conquer, as you said.**

**Another thing you mentioned was product and user orientation, meaning thinking, “How does the thing I'm working on contribute to the goals of the company, to user value?” I think these are the two things you mentioned, which you can invest in and help you when you switch careers. Anything else that you think is helpful?**

40:43

Chris

I try to avoid giving general advice, because what I learned is – different people have different career paths. Before this interview, I listened to a number of other interviews from DataTalks.Club. I noticed one thing that people arrive, especially machine learning, from very different directions. It's a very new industry, and we have different experiences. But maybe there was one general advice I would give. Especially when you are trying to build this T shape and get one stem where you go deeper, try to find something that you are genuinely passionate about as an individual. This will give you a great advantage if you find at least one area that you're just genuinely interested in.

I found out that people who just enjoy the process of learning something, going deeper in that one area, will always be ahead of people who are just focused on results. If you just want to get a job or finish a project, and you hate the way of getting there, you will be at a long term disadvantage compared to a person who just enjoys, let's say, the process of writing IAM policies. I remember I listened to an interview when the person you talked to said that he relaxes by writing these policies. So I'm pretty sure he will be way better at doing that than someone who doesn't like it. If you, as a junior, find one thing that is really exciting to do, try to go really deep into that – develop expertise. And for other things that you're less excited about, just do them as and learn them as you need them for your project.

42:26

Alexey

**It was the interview with Tomasz, which was about his transitioning from Data Science to DataOps. That's a very cool interview. Yes, not many people enjoy IAM roles. and then debugging them was a nightmare. I don't know how people can enjoy that. Luckily, we have people who do enjoy this, so they can help with this stuff. I guess, the root cause of many problems, when it comes to cloud and AWS, are these improperly configured IAM roles.**

42:59

Chris

Yes, it's very common. So I think this is Tomasz’s superpower that he genuinely enjoys this.

43:06

Alexey

**Would you suggest buying a rubber duck?**

43:12

Chris

Like a physical one? [Alexey confirms] That's a good question. I don't have one. I also use my colleagues for that purpose. This is another general piece of advice. If you get stuck with something, just talk to a teammate and it usually helps. Especially now, when we work remotely, I think I would actively recommend against it and I would rather say – talk to someone. Now that you mentioned this, I realized that ChatGPT could also serve as a rubber duck. For example, I started mentoring sessions with a person who is just learning programming. And at the very first meeting, I recommended Chat GPT to her and to use it as this assistant when she just has a question. We still have sessions between my mentee and myself, but in between, she's using ChatGPT and she's very impressed how well it can work as a secondary mentor. Maybe this is a better, upgraded rubber duck.

44:25

Alexey

**One of my ex-colleagues just bought a pack of rubber ducks and put one on everyone's desk. We still have quite a few of them around the office. Even though there is a duck near me, I don't really talk to it. Because I think it would be weird. Right? I don't know. I just don't feel like talking to a duck. [chuckles]**

44:46

Chris

Especially in the office, I think it can raise some eyebrows. [Alexey laughs]

# Tips to enjoy learning new things

44:52

Alexey

**One thing you mentioned is that if you enjoy the process of learning, you'll probably be more successful because you will also enjoy doing this thing. I wonder – what can I do to enjoy the process of learning? Are there tools, tips, tricks that can help me there?**

45:15

Chris

That's a very good question that I also thought about myself. Sometimes it just feels like magic – you either have it or you don't. It's difficult to force yourself into liking something. I think in the beginning, especially, it helps to just let your curiosity drive you. If you see something that is just interesting to you, try to go deeper. I think now there are quite a lot of things that are out there that could be interesting. For example, for me, such a thing was image generation with these diffusion models. In the end, I had to stop because I'm not going to become a data scientist, and I'm not going to get a PhD in that area. But I got relatively deep, given what I’m supposed to do, just because I was really excited about the results and how it works.

Let your curiosity drive in the beginning. That's one piece of advice. Another one that I noticed is that sometimes, if you just force yourself for, let's say, even half an hour or one hour to look into something – basically, just start. Often you discover that “Oh, actually, it is interesting. And I would like to learn more about it.” So just starting is good advice. Some practical tip for that, which works for me is the Pomodoro Technique. If you just know that, “Okay, I will have to work for 25 minutes, then I will have a break. It's not so painful.” Then after 25 minutes, you discover that this is really fun and you just keep going.

Maybe the last thing that I noticed worked for me – I like to listen to podcasts and interviews with inspiring people. For example, when I listened to a few episodes of DataTalks.Club – and I just chose them based on the titles of what interests me – I felt that afterwards, “Yes! I really want to go deeper into this.” Because there's a lot of enthusiasm from people that you had on your podcast. I'm not sure if I can recommend another podcast on this one – your competition. [chuckles] I'm a huge fan of Lex Fridman’s podcast. He started as an AI engineer at MIT, and his podcast was mostly about AI. Now he's someone who’s ventured into very different areas and he talks to people who are not necessarily working in technology.

I recommend going back to his earlier episodes, if you're interested in machine learning, because he interviewed very famous and accomplished people in the field. Just listening to them can broaden your horizons and can inspire you. I would say that inspiration is just part of it. In the end, you have to just sit down and do something yourself, because you can have all the inspiration, all the podcasts, but what matters in the end is that you face some of the pain yourself and try to do something yourself.

48:23

Alexey

**When you say “just start”. Okay – I have my Pomodoro. I put it on. The clock is ticking. What should I do next? For me, sometimes it's like I'm staring at a blank screen – “Ah! What do I do now?” [chuckles]**

48:37

Chris

Okay. A while ago, I think it was harder, but for the last few months, you can just ask ChatGPT.

48:43

Alexey

Yeah, I knew you would say that. [laughs] I mean, I literally do that too now. For example, recently I was working on a proposal for a PyData conference that we will have in Berlin soon. And you know what I did at the end? Just talked to the ChatGPT. Yeah, that helped. People are probably tired of hearing about ChatGPT by now. In general, there's so much buzz on social media.

49:12

Chris

There’s also another answer to that without ChatGPT. If you get stuck, I think it's a general skill that's very important for software engineers, for people working in tech – to break down the bigger problem into smaller ones. If you have to build, let's say, a data pipeline classifier – just try to break it down into smaller things to the point where you know how to implement it. If you don't, just keep breaking it down.

49:44

Alexey

**But what if it's a new area?**

49:47

Chris

People have different learning styles. For example, now there are a lot of really cool tutorials on YouTube, if that's your style. Some people prefer reading. Reading, I think, is a bit faster and more efficient, but maybe more involved. Just try to get anything on the screen – even one line of Python in a notebook or your script, and then usually, you know what needs to be done next. Starting is always the hardest part.

# Recruiting generalists

50:22

Alexey

**Right. When we were talking about the cons of being a generalist, you mentioned one thing that recruiting might be a problem. I was thinking, why is that? From what I see, sometimes, when somebody leaves a company, the hiring manager (the manager of that person) wants to hire the exact same person with the exact same skills. What they do is come up with a job description that’s just “Okay, what was this person doing? Let's just put all that there. That person had five years of Spark experience – let's just go with the same thing.” Then at the end, we see a job description that looks exactly like what this person was doing. And then, of course, it's not possible to hire a clone of that person.**

**Eventually hiring managers compromise, “Okay, they don't know Spark? Whatever, we will help them learn this.” Do you think that's the only reason why we have these job descriptions that look like a super expert? Or are there other reasons?**

51:33

Chris

I have to say that at Zalando, I'm quite fortunate because people who are writing these job descriptions are usually aware of this problem. At Zalando, we change quite often, and it's expected that people will learn different things. We try to avoid being too narrow. Most of the job descriptions I have seen are sufficiently flexible. Where is this coming from? I hope it will disappear. Technology is changing so fast that you just shouldn't focus on something that is too narrow.

If you want something super narrow, maybe you don't need a full time employee – just a consultant to solve one specific problem. But if you're investing in the person for longer periods of time, it's more important that this person is flexible, can put something together and build it rather than having just one specific narrow set of skills.

52:30

Alexey

**Then it doesn't look like a big disadvantage, because there are companies who realize this and try to put rather broad job descriptions, like Zalando. [Chris agrees] If I'm a generalist and I want to apply to a job position that looks like an expert job description, should I just apply and let them decide?**

52:52

Chris

Yes. Just try the interview. Of course, it depends. Sometimes it is understandable when the company wants an expert. If you’re working on, maybe not using some existing model, but you want to build a new machine learning model, maybe a PhD in the relevant area is actually someone you want. But then, of course, the cost is that it could be harder to hire the right person.

53:20

Alexey

**I'm thinking about an example from OLX. When we were looking for a person to join our recommender team, we were building a new team from scratch for that. We thought, “Okay, if we want to hire somebody senior for that team, that person should have experience with recommender systems. Maybe other things are not a must, but this thing is a must.” I guess you need this T-shaped profile, where the T (the stem) is the recommender experience. So it probably depends on the needs of the team. Sometimes you just need a particular set of skills, and they are a must.**

54:01

Chris

Yes. Well, I think it's getting a bit easier in the area of ML. Even just three years ago, it was such a new field that it was harder to find experts. But now, let's say you want someone with recommender system experience, it will be easier to find such a person than just a few years ago.

# The job market for generalists vs for specialists

54:23

Alexey

**In general, do you think companies are looking more for generalists or for experts? From what I hear from you, Zalando is broadening the job descriptions. And from what I see in terms of what we do at OLX, we also don't usually hire for a super niche set of skills. What do you think, in general?**

54:47

Chris

It really depends. Sometimes the companies have very specific objectives to achieve and they need experts. Zalando is also sometimes in need of specific expertise. Maybe it's also my personal survival bias that I tend to gravitate towards roles where they require a broader set of skills, because that's what I can bring to the table. I wouldn't worry about this if I was on the job market.

Just try to develop something that you like. And then develop this horizontal bar the letter T to look slightly around your core area of expertise. We're still, in general, as a tech industry in a lucky situation where there is still a lot of need for people, especially in the field of data and machine learning. So focus on something that is interesting to you.

55:46

Alexey

**Which today, I guess, sounds easy, but sometimes there are so many things. What is interesting for me if everything is interesting?**

55:56

Chris

This is something I struggle with as well. You also have to learn. This is the opposite from what we were talking about. You also have to say no to a lot of things.

# Narrowing down your interests

56:06

Alexey

**Yeah. Like coming back to our discussion about your sabbatical plans, right?**

56:10

Chris

Yes, I definitely have too many interests for the limited time I have.

56:15

Alexey

**How do you deal with this? Do you have any suggestions? What works for you?**

56:22

Chris

What I do is – I'm a huge fan of to-do lists. If you just write down everything that is on your mind – just try to capture it. And then you can prioritize. Then you can also just drop some things off the list.

56:37

Alexey

**To me, sometimes to-do lists feel like procrastination. Like I'm not really doing anything – I'm just doing a to-do list, instead of doing the thing. Do you get this feeling?**

56:50

Chris

It could happen. But I think it's even worse when you don't write it down and it's just this general mess in your head – when there are a hundred different things and you don't know where to start. When it's written down, at least you know, “Okay, I can do this.” And the other things will go down in priority.

57:09

Alexey

**I'm not sure if it's related, but what I noticed helps me is having some sort of deadlines. Also, doing some sort of project work. Instead of, “Let's figure out how Stable Diffusion works,” it would be “Let's figure out how I can use it for solving a specific problem.” Maybe for Stable Diffusion, it's a bad example. But usually, there is a new piece of technology I want to try and I give myself a couple of days. Then it kind of helps to stay focused on this specific thing.**

57:46

Chris

I think deadlines definitely help. There is this saying that “If you have a software engineering task, it will stretch to the time you allocate to it.” Sometimes if you allocate less time, it will force you to focus and say no to other things.

# Chris’s book recommendations

58:04

Alexey

**Do you have any book or other resources that you want to recommend to the listeners?**

58:11

Chris

Um, I thought about this. Right now, I noticed that, especially in the last 10 years or maybe even less, with the advent of deep learning and machine learning, it's harder to learn from books because things are moving too fast. In my personal experience as a software engineer, some time ago, there were books that were general, for example, Clean Code was definitely an influential book. Pragmatic Programmer is another classic. Recently, there was a second edition released that is more updated. I would recommend this one. These books teach you skills that are transferable to other areas. However, they are mostly aimed at software engineers, so not everyone from our audience will be super focused on this area.

Right now, I would say I learned more from YouTube and tutorials and just finding things as I need them. A colleague of mine, when I was new to Amazon SageMaker recommended just going through the documentation. I'm not sure if it's always a good idea. I downloaded the PDF. It's 3000 pages long. Here I would recommend just looking things up as you need them. I don't want to sound very boring but ChatGPT is, again, an amazing resource to poke and get general ideas, especially when you get started and you don't know when to write the first line of code.

59:53

# Conclusion

Alexey

**Do you have a favorite YouTube channel?**

59:58

Chris

With programming, not really.

60:04

Alexey

**Outside of tech, maybe?**

60:07

Chris

Outside of tech? It would be completely outside. My other hobby is photography. So there are some photographic channels that I like. But they’re pretty far from what we’re discussing here.

60:24

Alexey

**Well, if you’re interested, maybe give us a link and we’ll put this in the show notes. If anyone is interested in photography too.**

61:31

Chris

Sure.

60:32

Alexey

**Okay. That was fun. Thanks for joining us today. And thanks, everyone, for joining us and listening to our conversation. That was a lot of fun. Thanks, Krysztof, for sharing all your experience with us, for giving us advice. Have a great weekend, I guess.**

60:52

Chris

Thank you for having me.