2:02

Alexey

**This week, we'll talk about machine learning and data at enterprises. We have a special guest today, Alexander. Alexander is responsible for data and artificial intelligence at the consultancy, called Königsweg. I hope I pronounced it correctly with my German. I've lived in Germany for some time, so I think now I can pronounce umlaut.**

**You might know Alexander, not from his work Königsweg, but from his involvement in the PyData community. He's one of the organizers – or probably, I should say chairs – of the PyData conference. I think this is the biggest data conference in Berlin and my favorite conference. A couple of months ago, I was at this conference in Berlin and it was a really awesome conference. I actually invited a couple of people to this podcast from this conference, and Alexander is one of them. I'm very happy that you joined us today, so welcome.**

3:05

Alexander

Yeah, thanks for having me.

# Alexander’s background

3:08

Alexey

**Before we go into our main topic of doing machine learning at enterprises, let's start with your background. Can you tell us about your career journey so far?**

3:19

Alexander

Oh, it's a long story, actually. It actually starts right here [points to vinyl records on shelves] [chuckles] with the records. Actually, I used to study law. I hated IT in school, like Pascal – I thought it was super boring. I couldn't put use to it. So, actually, I was a music enthusiast in the 90s – I was DJing. Finally, I was co-owner of a record company. The company was actually doing pretty good. And there was no software around.

3:50

Alexey

**That’s why you have so many records in your background? [cross-talk]**

3:54

Alexander

Yes, actually, from those days. Yeah. I have a tendency to make stuff I'm passionate about and that I care about, also in my profession. It's very similar with data and programming and Python, which came later on. Actually, the record company did pretty well. We had administration problems and there was no software around, so I had to build it – or I decided to build it – so I taught myself programming with two books at that time. From there it goes on.

The record company didn't really survive the crisis at the turn of the millennium. But it wasn't too bad because I was moving into programming and soon, as a whole, everything about machine learning started. I was just super excited. I took all the courses I could get and decided to move on to the field of machine learning and AI.

4:56

Alexey

**Yeah. So now you work as a partner at Königsweg. What does it mean to be a partner and what do you do as a partner?**

5:07

Alexander

As a partner, of course, I'm responsible for my team – the team working on data and AI topics – to look for new people, of course, because Königsweg is growing. I always look for support in projects – if it's freelancers or people to increase our staff numbers. Of course, as a partner, I have to think about what our strategy is, which products and services we should focus on. It's a very broad field and we always try to narrow down. We have a tendency to work on cutting edge, state-of-the-art things, so just implementing boring stuff is probably not our thing.

Of course, it's very often part of a more exciting project, so this is my work and also being partner and founder of boutique consultancy. It’s fun because it gives you a lot of freedom to decide what you want to work on, which people you want to work with – and this is not only people we want to bring in our team, it’s also clients. We will not accept just any client. We say, “Okay, that's not a good company culture. They don't really share the same values.” So we wouldn't just do it for the money. We would just say, “Well, it's not a good fit, because we like to get things done and not change lifetime against money, because that's just a waste.”

# The role of Partner at Königsweg

6:50

Alexey

**I'm also curious about the word “partner”. I work at a product company, and we usually have a Chief Technical Officer or VP of Engineering, so “partner” is not something we use as a job role. This is, I guess, more common to consultancies and also to law firms – so people who do services for other companies. Here, a “partner” means that you're responsible for an area of the company. [cross-talk]**

7:26

Alexander

Basically, as a partner, you're driving a certain area in the company – you are responsible for that. And of course, with all the support from other partners. We collaborate on things all the time. It's like I'm running the department, but I can decide where I want to move, so there's nobody [I really answer to], except we meet as partners and discuss things that we should do, but nobody can tell me if we should focus on MLOps, as we are doing now, or if we should implement some other things. This is basically my call to make.

8:11

Alexey

**Basically, the company consists of many independent units, and each unit is led by a partner, right? You're a leader of your particular unit.**

8:23

Alexander

Yeah, basically. But there aren’t too many – there’s five partners. We are currently growing, and we expect to really grow in the field of data and AI and financial services. But for us, it's very important to work on stuff we care about as well. There's a lot of stuff you could do, but it's boring. [chuckles]

# Being part of the data and AI community

8:50

Alexey

**I imagine that this is not simple work – not easy work, right? You need to do a lot of stuff: you need to talk to clients, you need to think about what the strategy should be, what you should focus on. Then you also have people who report to you. That's what you do as a managing partner of the company, but that's just one thing.**

**I also know that you do a lot of other things outside of your work. You're a chair at PyCon DE & PyData Berlin conference, right? You're doing many other things. I also checked your GitHub profile (I can link it). I went there, I checked, and I saw that it's quite an active profile. So how do you manage to do all that? You’re a managing partner, community leader… [cross-talk]**

9:30

Alexander

I don't have other hobbies. [laughs]

9:35

Alexey

**That’s your hobby, right? [chuckles]**

9:36

Alexander

I have a tendency to combine things I care about with my work. The community is part of that. I enjoy the community a lot, especially in the Python community, there’s a lot of input. A lot of the Python community has a very, very healthy culture. It's also quite broad from the topics you can find. You can find astronomers, you can find web developers, and I think this is a very, very good mix. Because I believe in learning across domains – it’s a very good thing to see people who work on very different topics and to learn, “Hey, you have a similar problem!”

For example, if you have image problems, talk to astronomers. Because they have real image problems. [laughs] And they solved them already – they probably solved them with way less hardware than other people, and they throw it in the cloud. I think it's always important to keep the exchange going. Of course, there's always a lot of interest from academia, but academia can also learn from business or business experiences. I just like to go to conferences, talk to people, and it's always very enlightening and joyful. I like the atmosphere that creates the “there are no stupid questions” culture. This is one thing I preach everywhere – also to clients – there are no stupid questions.

# How Alexander became chair at PyData

11:13

Alexey

**How did you become a chair? You mentioned that when you became interested in machine learning, you took all the questions about this you found, and I guess at the same time you started to look for local meetups, local communities, and you started attending all these meetups. This is how you became a chair, right? [cross-talk]**

11:33

Alexander

Actually, it was like an accident. [chuckles] My first PyCon was in Cologne 2013. And this was my first Python community experience. And I decided to go to EuroPython in Berlin, just across in the BCC again, where we have the conference now. I heard about the European society and I said, “Okay, let's go there. It's free.” And then they were looking for an auditor and I said, “Yeah, okay. Auditing books.” EuroPython, at that time, was to audit like 10 invoices a year. [chuckles] The conference was still run by the local community. So it wasn't actually a lot of work. I was like “Okay, I can do it. If nobody else wants to. I can do it myself.”

I've interacted with a lot of people I've never met in my life via email, and we were building the Bilbao conference. Then basically when I came to Bilbao, it was the first time I met, like Alex Savio, Christian Bara and many others who have been working on the Bilbao Conference. If you had asked me beforehand, “Can you organize like a 1200 person conference? In Bilbao? By email? Remotely?” [chuckles] I would have said “No way.” Actually, it worked pretty well – it was a great conference, it really worked. The Bilbao conference was a reset for EuroPython to run in a different fashion. Basically, everything was just reorganizing. Actually, while we were building the process, I ended up being program chair with Alex Savio. From there it goes.

I wouldn’t really say that I like to organize but I think making things happen is very powerful. Of course a conference gives you a lot of room for creativity to try new things out. You also see that it resonates with the community – you help people. You also help people who are not at the conference by recording the videos and getting everything together. I quite liked that, so I just stayed on EuroPython, and from there just went on EuroPython, where Sebastian and Peter from Karlsruhe nearby came and said, “Oh, we should bring PyCon to Karlsruhe.” And it's like a very connected area here, so we run PyData southwest, in Mannheim, Heidelberg, Kaiserslautern, and nowadays, and so it was very close and they said, “Hey, do you want to help? You have the experience from EuroPython.” And I said, “Yeah, of course.” [chuckles] This is how I became part of the German conference.

Actually I was just helping a little bit with EuroSciPy and then the Program Chair left and then Valeria just wrote in Telegram “Alex should do it” and I said “Yeah, why not? So that's why I ended up being involved in many conferences. But it basically was not in my plans until I die. I'm more thinking about how I can *not* be involved in so many conferences? I think we also have to consider – we need to make room for fresh blood, fresh ideas. I'm currently just working on the European Summit for organizers in November because, of course, organizing conferences and everything is a lot of work and on very few shoulders. I've been investigating different people who have figured out how to run a conference, not being core organizers, that haven’t been basically burned out. And I haven't found one yet. If somebody from the audience knows one, please contact me, because we need to find those structures.

The community is growing. The conferences are growing. The experience we want to provide to the community is growing. So we need to rethink how we organize and work on conferences, because it depends on too few people. We have to work on that. So that's why we at EuroPython are organizing a summit in November and we will invite organizers from all over Europe and also discuss how we can help each other by mentoring and standardizing some processes. Because very often in the conferences, in local communities, or new people have to reinvent the wheel for the fifth time. So maybe we can help that because it's a problem.

16:25

Alexey

**So if anyone is interested, please contact Alexander.**

16:28

Alexander

Yes, if you want to help at conferences – contact me. [chuckles]

# Alexander’s many talks and advice on giving them

16:31

Alexey

**Who knows, maybe you will become the next chair of the PyData conference in Berlin, right? [chuckles] Anyways, I also noticed when I was doing a little bit of research that you're a very active speaker. So you speak at all these events that you organize – maybe not on all of them, but your talks appear often. In Google, you can put in a name and then look for videos, (there is a special tab) and when I did this with your name, I found 604 results. Did you actually do 604 talks?**

17: 10

Alexander

No, actually, I think it's probably playlists or something else. Actually, I think it's a lot. I have stopped counting, but I think it should be something like 100. I am planning to recount soon, because maybe the hundredth talk should be something a little more special. Yeah, it just adds up during the years. [chuckles]

17:36

Alexey

**None of what I’m asking right now is what we prepared, but I'm really curious. I find problems sometimes getting inspiration, like “Okay, I probably have done something interesting. But how do I package this as a talk? How do I find a way to share this? How do I find material to talk about?” This is a problem for me. For you, with 100 talks, you’ve probably found a way to generate talk proposals. Maybe you can share it?**

18:09

Alexander

No. Not at all, actually. Every time I propose a talk and when I finally have to prepare it, I hate myself. Like “Why am I doing this to myself? I lack time.” Of course, for me, it's important to develop and deliver quality talks. I don't just want to say “Okay, this is a talk and it's some topic that people have seen before.” Of course, I can narrow it down to a very simple formula. If you have something to say, you don't have to worry about what you will talk about. A lot of the things, “What should I talk about? Is it good enough?” Which is what a lot of people think – it's just in your head. I'm sure you have like, five topics that you can talk about and share your insights with the community. It doesn't have to be the latest hype or tech.

I just discuss things I learned when I do talks about something new. And it's not only me. I’ve been program chair to so many conferences, and with so many speakers – of course, you become friends. Then we also talk with one another, like “Hey, what about you?” I realized many people will say, “Okay, these are great speakers, actually.” Very often they talk about things they just learned. Doing the talk and delivering the talk is part of the learning phase. So it's not like you're an expert and then you are eligible to talk. I think you have more value. And this is also what I'm trying to point out no matter where I am – if it's the community or with customers – don't talk about the shiny things. They're nice. Yeah, you can talk about them. But don't forget to talk about the mistakes you made and how you solved them and where the problems were and where you were stuck, because this is where we can learn from each other. We hardly learn from impressing each other with cool stuff.

20:12

Alexey

**So a top proposal could be something that you don't know yet, but you want to learn – then you've come up with a proposal and by the time the conference happens, you have to actually learn this thing to talk about it.**

20:25

Alexander

Yeah, it's not that short notice, so you probably have some time. When I did the Pandas talk, I was already working a lot with Pandas, but I did deep dives by providing Pandas talks and also like thinking about “Okay, what should we basically point to for Pandas at that time? Because all the Pandas tutorials were like – nobody was explaining the index at that time.” And I think if you understand the index [cross-talk]

20:51

Alexey

**Nobody understands the index except maybe the creator. [laughs]**

20:56

Alexander

It's an important topic to say, “Hey, there's the index. It's a very important structure when you work with the data. Actually, you can do really cool things for that as well, which are really useful and big timesavers.” That was then. Then, I did a talk series about Deep Learning and AI, which was “Deep Learning for Fun and Profit” – taking blog posts and trying to do style transfer, text generation, speech generation and other things. It was quite fun, actually. I miss that a bit because I learned a lot about deep learning. I learned a lot about how to approach it.

Of course, being a partner at a consultancy, I also have to consider “Oh, there might also be customers there.” So it can be toyful, but it's also “What's the connection? What can you learn as a business from that?” The learning from the Deep Learning for Fun and Profit series is basically “Yeah, you don't predict deep learning. You have to experiment. You have to be free to experiment and you cannot basically decide if something's going to fly or not.” That was quite useful. So, yeah. Stuff like that.

22:09

Alexey

**The range of topics – I think you mentioned Deep Learning for Fun and Profit [cross-talk]**

22:16

Alexander

Yeah, Deep Learning for Fun and Profit. That was quite enjoyable and that was fun. Of course, it was many weekends spent on making things happen. [chuckles] But it was good, because for me – many people think a partner at the consultancies is just like “Oh, yeah. You're just like a manager?” And I said, “No. Yeah, of course, I'm also a manager – but I'm a generalist.” Also if I'm with clients, it's very important for me to understand the tech. We’re discussing, we're working on it – so I'm still hands on. I’m not just on this level “Please, team – explain this new technology.”

No, I am probably not the expert on everything, but I have always worked with everything and I can give very good insights on what's useful and what's not useful. So that was the deep learning part. There are so many misunderstandings and hype about data and AI – and why they do not work at companies, a bit small or bigger companies. But for example, at EuroSciPy I will talk about software engineering, because I see that there's still too many data scientists around who have not heard about it and EuroSciPy academic conference, and students work with Jupyter notebooks – if we really want to deliver something reproducible and stable, I thought about now it's time to give a talk about software engineering, because they should really know about it.

# Explaining AI to managers

23:48

Alexey

**Your talks are quite diverse. You also have a talk about Pandas, you mentioned MongoDB. One of the talks – or variation of a talk – I noticed in quite a few talks that you had that the topic is “explaining AI to managers.” This is actually the topic of today's interview. I guess you, as a manager and as a managing partner, need to do this quite a lot, especially when you talk to your clients. Then you probably talk to people who do not necessarily have a lot of knowledge about machine learning, so then you have to explain all these things. This is how this theme appeared, right? This is how you started talking about this, yeah?**

24:31

Alexander

Actually giving talks was really helpful to explain things to other humans. When I started doing talks, I was really bad at explaining – because I was too detailed, I nested sentences and everything. Very often people said “Alex, I cannot follow you.” The talks also helped me to evolve personally – to simplify, sometimes – because if you work on software and you want to deliver quality and you have an engineer’s mindset, the downside is you want to be very exact. But sometimes simplification is not exact enough – sometimes you have to simplify things to get the message across. This helped me a lot because the clients we work with are looking for state-of-the-art tech, they work with open source, and very often we also have to explain things.

Proprietary software that you buy is *probably* a solution. It depends on what you want to use it for. But data and AI is something you have to build, you cannot just buy it. It's not like a piece of software. Very often, we also support people in the company to convince decision makers because very many people in management think of data and AI as “Yeah, it's just like a piece of software you buy and you hire people to implement.” But it's not. Data and AI – it's two things. Getting your data in order so you actually can use it – and if a company is older than five years old, the data is always messy and distributed. You have to really work on how you organize your data, how you can make it accessible, and find new approaches for more AI-centric data.

On the other hand, of course, there is sometimes a software tool that can help or help for a part, and it's probably easier because not everybody has tons of skilled developers. That's the other thing. So, our part is very often to help make the right calls. Then if it's implementation in open source, we are there to make it happen. We're not dogmatic about open source, but I'm a strong believer in it – everything in analytics, prediction, data – basically why not use open source? It's way easier, because if you buy software tools, you also have to learn it. What I like to see is that many decision makers don't want to be dependent on a piece of software from a supplier – that is probably a startup – they don't know if it’s gonna last. Or even with bigger software vendors – they invent new products and two years later, they're gone. So we say again, “Okay, just enable yourself. Use open source. Build up a team. Build up your team’s skills. Start working with the community. Contribute to the community. And then you basically have all the freedom you want.” Of course, it's work. But if you miss a piece – if you miss a feature – you can just go ahead and implement it.

# Why being able to explain machine learning to managers is important

27:53

Alexey

**That's the beauty of open source, right? This topic of explaining machine learning to managers – for you as a consultant, why is it important? Can you not just tell your clients “Trust me, it works. It's a bunch of math, you will not understand it anyway. Why should I bother explaining it? Trust me, it works.”**

28:17

Alexander

People have to make decisions, of course. I mean, we were talking about building software. In a larger team, it takes a budget. And of course, other people need to know, “Why should I spend the budget? Is it the right call? As we always say, “Nobody was ever fired for buying something from Microsoft.” [chuckles] This is the other thing. Of course, here is the IBM solution and people say, “Okay, let's buy this.” And it's very often the right call – but very often, we still have to explain “It's not software.” We build software to work with data, to build models, but it's not a software project. Many people are not fully aware of that yet. It’s getting better and better.

I would say, like five years ago, many C-level people were not aware that open source or a data scientist is something different than somebody who sets up your email, which is just like a configuration – We're a dashboard and because it's all IT, they don't understand the difference. Of course, it's also nice to see the generation changes happening. More and more decision makers, younger decision makers, enter the field, many of them know Python, because of studying, writing their PhD thesis, stuff like that. So it's getting better and better. But still, we need to support them – which open source tools should they use? There's a new framework every day. What's the choice?

Yeah, it's cool. But on the other hand, you always have to see “Okay. How long is it going to last?” We lack resources – everybody likes resources, so we have to help them to make the right calls and go for something stable, which can work for multiple years. This is also very helpful, being around conferences and interacting with the community – I can have a broad field of tools that are around there and I can get good advice on which is good, which is probably even better but with a smaller community and so probably it's better to use the other tools.

30:42

Alexey

**If I understood you correctly, the main thing you want to solve when explaining AI to managers is not to explain how actually the latest transformer models work, right? What you want to explain is that, in addition to this model that does this magical thing, there are many, many things around this that you also need to think about, like all the things you mentioned – it's not just data and AI in one box. There is data, which is like a vast thing, and then there is the AI component, which sits in the center. That's what we need to explain. Right?**

31:18

Alexander

Actually when I need to explain – if you're serious, the first question you have to ask is “Is this really in the strategy of your organization? Or is it something like “Oh, yea – we should try this. I've read it in the newspaper or seen something on LinkedIn.”?” I think the times of “oh, we should try this and maybe…” are definitely over now. Even if it's in your strategy or not – and if it's not, good luck. [chuckles] Other companies will eventually be faster and better. So if it's in the strategy – people are fascinated by AI. They want to know about neural networks, artificial intelligence, and, of course, there’s a lot of science fiction ideas, hype in their head. Very often, I'm just going to explain to them, “Okay, you actually don't have to worry about building a neural network *at all*. There’s something that you can just get from the shelf. There's so much research on neural networks.”

So maybe you have to make a choice, but you never have to design your own neural network. Very few companies need to do that, but then they're super-specialized AI companies. Basically, the magic is already out there, you just have to scout it. We can help you make the right calls. The challenge is establishing the right company culture. I always say “Get your data right, to scale experiments.” Because you can never know where the real value will be for the company. Many things are good ideas, but I also have to explain, for example, Google. Google does a lot of research with ML and deep learning. There are also the things that work in research – some of them go to production, and even then only 5% survive there. Things which might sound plausible will probably not be the solution and things which you don't think about it all might be great solutions.

You have to establish an openness – you need to establish culture, to get the data, to do collaboration, to openly discuss problems. Of course, it's hard. You do something fancy with machine learning and who wants to do a presentation “Yeah, sorry. It didn't work out.”? But we very often have this in projects. We say, “Okay, we're looking for a solution.” For example, a client of ours was looking for a solution in natural language processing. They had like 30 years of research data and documents there, and they said, “Okay, yeah. Keyword search is not good enough. So what about building a knowledge graph?” We said, “Yeah. Well, of course. Let's do this.” And then we started building the knowledge graph. But then we had to say, “Sorry, it's 30 years of research, but it's still not good enough. The knowledge graph is not building up.” Of course, you can get data from outside, but we always have to look for “What is the *real* problem we want to solve? What do you *really* want to accomplish?” And here it was, “Okay, we need more insights. We need better access to this corpus.” Knowledge graph was not the solution, so we don’t tune things and do a knowledge graph and have a nice presentation and then say “Yeah, bye-bye. Good luck with the knowledge graph.”

We didn't believe that this would ever fly into something useful so we said “Okay, we have alternatives.” Here the alternative was keyword extractions, finding entities, summaries, and actually, clustering. We put this into a very nice UI and then they really had a very good thing and could get a totally different perspective on all their research documents. They could find out “Okay, what was the better recent research at the time?” For them, it's not just “We want to look at this because we like to dig in our history.” We're talking about a research department. Research departments have like a billion figures to work with. So, of course, if they know, “We researched this 20 years ago and it didn't work,” or “We know it didn't work, but why also it didn't work.” It can help to save a lot of money and resources. Or they could reinvent the wheel and probably come to the same conclusion. Of course, you have to be very open to – if there are problems, you have to be very open and transparent with the clients and not just trying to work on the happy path. [chuckles]

36:07

Alexander

Of course, I'm happy to say we always ended up with the happy path. We never had like a total failure – we always found a good solution that addressed the problem. But, in between, it's quite a ride and you have to explain things, because also, there's very often experts involved. When we did that project, that was still a thing. “Oh, can’t we use BIRD.” “Yeah we can use BIRD, but we don't really think it will help solving the problem that we’re currently addressing.” We're not paid to play with the newest tech. We are paid to develop value and help people save time, to be more effective, to make better decisions.

# The experimentational nature of AI and why it’s not a cure-all

36:50

Alexey

**Would you say your biggest challenge in explaining AI to managers is conveying the experimentational nature of all these projects, saying that, “Yes, there is this cool tech that you heard about from social media, but it might not be the solution for your problem. We need to experiment, we need to play with different tools, and we need to have a proper way of evaluating if something is working or not.” Would this be the main challenge or not?**

37:22

Alexander

Actually, I used to hold back a little bit more in the past. I learned to be really upfront – that is the best thing. For example, we had a meeting with a client and somebody told me, “Okay, this is the important input from Frankfurter Allgemeine.” It was about some AI camp somewhere, and I read the article and it was full of nice, idealistic ideas and there's stuff people read about, “Yes! AI finished Beethoven's unfinished 10th Symphony! I’m really excited about this.” What is my answer to that? I said, “Hm. I'm very sorry. But we have to accept that the 10th Symphony of Beethoven will never be finished because of a very simple reason – Beethoven is dead and he never finished it.” I think Beethoven is a very good example, because if you just go one symphony back and say, “What if the 9th Symphony was unfinished?” We only had eight. Of course, the unfinished symphony could be something that sounds like the symphonies he has composed before.

But, especially if you look at the ninth symphony of Beethoven – there’s a choir, which is the European hymn and everything. This is invention and AI at the current stage doesn't really invent things like that. It can be very good at repeating things it has learned, of course. This is a very good example explaining it with Beethoven, because everybody gets excited and thinks, “Oh, this is such a great thing.” “No, no. You just get the same thing we know already. This is a strong suit, but I don't think it's a good solution.” Also, I think it’s a little bit disrespectful. [chuckles] Because Beethoven no longer has the ability to say, “No, this is not what I intended.” So stuff like that. I say be very, very upfront. But, of course, it's very important to be respectful about it and, of course, there is a lot of hype – there's a lot of startups around who claim things. It’s also really good to be connected to the community – because one CEO of a startup was also cited in the newspaper and they said, “Yeah, yeah – we’re already there.” Basically how he acted because they cited him was like “Yeah, we’re already there.” And I said, “Hm. Well, actually, the person that wrote the research paper gave a talk at a meetup and, actually, the expert working on the topic gave us a little different version when delivering the talk about the majority of the tech.”

Although the tech is still exciting and it's very good, I can also give good examples why it's not just hype and why I'm not just saying “Oh, don't believe the hype.” I can also bring evidence and narrow it down and say, “Hey, what do you really want to solve?” I think the biggest issue is still company culture, like having domain experts and technical people and engineering teams work together on the same level, as one team. With bigger enterprises, there are departments – they have a requirement, they write a ticket or a user story, and then they just throw it over. Our message is “No, no. Work in hybrid teams. Work on this together.” Because we don't have time for paperwork and all this miscommunication in between.

Of course, humans are not really good at changing routines. We are routine animals. So I think the biggest challenge – even a bigger challenge than solving AI – is actually changing human routines. For me, one thing that absolutely belongs together – you won't be able to invent technology if you are not able to reinvent your company culture, or if you already have. For example, many startups – they start, they have this form that’s basically built-in from history. They're young, but especially big, larger enterprises, they go back decades. And of course, there are multiple generations of people working there. Of course, it takes time.

# Innovation requires patience

42:00

Alexey

**In preparation for this interview, I asked you, “Hey, can you think about some questions that I should ask you?” And you kind of came up with a few. We already talked about them and you added a few points. I'll just read these points. First one – innovation requires culture. And then the second – innovation requires patience. I think we covered the culture part. We talked about experimentation, being data driven, you shouldn't just chase the latest trends from Twitter or whatever. But the other thing, this “innovation requires patience” – what do you mean here? Why do we need patience? Will AI not just magically solve our problems tomorrow?**

42:48

Alexander

[chuckles] No. Because I think patience is very good to make good calls. I’ll give you another example. It's probably a little bit hard to explain without slides. I gave a more extensive talk about it at PyData London and it's on YouTube already. It has the same title as our session today. Basically, why patience [is important]. It’s actually quite interesting, because before this talk, I was complaining, “Oh. I hardly have any questions from the audience.” Probably I answered them already. I was actually making fun with Alessandro Salcedo about this, because you have to send your experience. Basically, this was one where we really started a conversation after the talk. I was pointing out, Hey, I like agile. I like retrospectives. Of course, there's a lot of scrum rituals.” But what happens?

For example, every three weeks, you do a retrospective. We could ask ourselves, or I think “What happens if you ask engineers and developers for problems? Will they ever say there is no problem at all? I don't know. Ask for problems and you will always get at least five problems.” Because we are problem solvers and, of course, we always have problems at hand. If you do retrospectives, for example, everything might be in order. But if you ask, “Okay, what can we improve?” There will always be tons of ideas to improve. But why not say “Everything's in order. We accomplished the sprint goals. We’re on track. Everybody's happy. We found a good working rhythm. Okay, this is the retro. Let's finish after 15 minutes, go for coffee or pick up work (because everybody's always busy).”?

Of course, we try to find more problems and try to over-engineer the whole thing as well. What I also learned by working with non-technical people – over-engineering is actually not only engineers’ problem. Many people do that. Even in management, they try to over engineer and ask too often, “Is there something we should improve? Is there something we should improve?” I say, “No, no. Patience.” You just need to get a more of a bird's eye view because things take time. It was quite interesting, especially the retros and agile for data, and data science – it quite resonated with the audience. It was a great conversation afterwards. [chuckles]

45:23

Alexey

**Interesting. I'm just taking time to somehow distill the main message here. We, as humans, always want to look for ways to improve what we currently have. Let's say we have a product – let's say we have search. The search is working fine, and we, as humans, if somebody comes to me and says, “Hey, what do you want to improve?” I can say, “Hm. Our search could be better.” And then it triggers a whole discussion of how we can make it better, even though maybe we don't really need to work on improving search right now. Maybe there's something else to work on. Right?**

46:03

Alexander

Yeah. Right. Or maybe just lean back and watch what happens. Because until something's happened, we already know, “Okay, we should improve this. We should revisit.” Maybe just take a step back, work on something else, and come back to it two-three weeks later and see, “Okay, the whole thing is now really not working anymore or is everything fine?” If you're very involved in working on something, of course, you see all the tiny bits that could be better. But are the tiny bits actually important for solving what you're working on? Sometimes I also fear that working on too many details might not be good because you lose the bigger perspective.

Maybe there are other factors, especially if you work in data and AI. Maybe you have solved the problem and you knew how to tune the algorithms even better. But maybe taking a step back could also be helpful. What about ethics and all the stuff? So lean back. Does this really work? Is there anything in the data? Take some different perspectives on working on the projects, and not just over-engineering the technical parts. In Germany, we have a saying, “Perfektion ist der Feind des Guten.” Or “Perfectionism is the enemy of the good.”

There's another thing I picked up from our software architecture book, “No big system will ever be perfect.” That's the nature of having complex and bigger systems – they will never be perfect. They *can* never be perfect. So we actually have to deal and work with “Okay, this is just probably good enough.” Especially in engineering, because as an engineer, you want to say “No, it’s not perfect.”

48:02

Alexey

**I think there’s a quote from Donald Knuth, or somebody else – from some famous engineer – “Premature optimization is the root of all evil.” Something like this.**

48:14

Alexander

Yeah. This is also something I also have to point out very often. We don't optimize for performance until we hit a bottleneck. Because I just know from experience, when you try to optimize up front, you will always optimize the wrong things anyway. [chuckles] So that's a huge part of communicating, being part of teams making data and AI happen, and also to say, “Stop here. Wait. We should do something else and refocus and give advice on that.”

# Convincing managers not to use AI or ML when there are better (simpler) solutions

48:54

Alexey

**Speaking of that, there is a question that we have. The question is, “Sometimes we don't need machine learning or artificial intelligence. How do we convince managers or business stakeholders not to use ML or AI when there are others who insist on using them?”**

49:10

Alexander

It's actually very simple. Usually, when I talk about it – let’s say we’re suggesting a reference architecture to our clients, so I say “If you're serious about data, you need a reference architecture, which is basically data in a data lake structure – data lake just being a concept, not really a storage thing, making data accessible. It will be accessible in a very consumer-friendly way, and, of course, with all the governance stuff being taken care of as well.” Even in first meetings, when people contact us about data, and I very often say already, “Okay, we have established this reference architecture, where you basically can get the data – it's really way easier to get the data. You don't have to research where the data is and pull this together.”

Because when we’re taking data projects, they often start from the wrong end, “Oh, we have this idea. Where can we find the data?” Yeah, you can do hundreds of these problems, but you will just build 100 zeros while doing it. So actually, if you're serious – get the data right, do the experiments, qualify – or not. Once the data's right and very accessible, 70% of the data used will likely be just business intelligence and analytics – no machine learning at all. Having the data accessible, building our dashboards for business users is the problem solved. Then maybe another 20% will be machine learning and if you're lucky, there’s like 10%, deep learning.

It depends, of course, on the domain and on the data mix. Most companies still have numerical data. If there is more unstructured data in the mix, or images, deep learning would be a bigger percentage. But I'm not just saying, “Oh, we only do data and AI.” “Okay, if we organize your data right, you can have better standard analytics. That's also a good thing. Business intelligence is not our enemy.” Our goal is to save people time, to help them to make better decisions faster, without handling Excel sheets or whatever all day.

51:45

Alexey

**Speaking of patience – I imagine this scenario where somebody comes and says, “Hey, we want to use these latest AI trends!” And then the reply is “How about your data pipelines? Do you have a lake?” And then you build the lake. And then “Can we use data science now? Can we use artificial intelligence now?” “No, no, no, wait. There are cases where we can solve them with analytics.”**

52:12

Alexander

I will say that the lake doesn't have to be full. If it's 10% or if there's some data in there, you can already start it. It's more about the right mindset, like, “Okay, get the data right and don’t look for nice machine learning or deep learning ideas.” And then you research data, because then we will likely have some export from the database and you’re basically just being disconnected from the system. Of course, if you really want to build efficiency, you have to bring it to production.

Then, of course, the solution for this is MLOps, like, “Hey, there's new data – retrain the models.” Close the circle, because there's also a lot of machine learning that machine learning engineers or data scientists also spend a lot of time on, like building models, releasing them, putting them somewhere. So MLOps is currently the best thing you can do to become effective and get things done. It’ll let you have people think about problems and not just waste time on stuff that you can automate.

# The role of MLOps in enterprises

53:27

Alexey

**So would you say that MLOps is the best recipe for machine learning in all these enterprises, or is it something else?**

53:34

Alexander

Yeah, definitely. Because it's not just because it's MLOps. MLOps is also… That’s another thing I always preach. It's not just programming – everything we do is also a standardization. We have to standardize things. We have to say to the whole company, “Let's work on this standard. This is how we do things.” If you just have a company, and there’s like 20 data scientists, and you don't say, “Okay, what's the common standard? How do you want to build CI/CD pipelines (and all this)?”

If you have 20 people, you probably have 30 different approaches to running CI/CD pipelines in the company. Then cleaning it up is basically impossible. Of course, this is also another patience part – don't think about, “Okay, how can we deliver things in quality and increase the quality rather than thinking about making many, many different approaches happen or not.” It's very, very important to standardize, to think “What's a good standard?” Also to question standards during the process until it's basically at the best level.

54:54

Alexey

**To be honest, MLOps lately feels like – because of all the buzz around this word – it feels like a buzzword, right? People will throw the word MLOps around everywhere, like “How do we solve it?” “Oh, with MLOps?” “What do we do here?” “Let's do MLOps.” I already imagine important people from McKinsey, all in suits, delivering a presentation in PowerPoint and then they have MLOps with big letters there, saying that it will solve all your problems. Do you also have this feeling that people just talk about MLOps without really knowing what it is?**

55:35

Alexander

I mean, we experienced the big data hype, “Everybody has to do Big Data.” “Why is your data not in order when you want to start deep learning projects now?” Of course there's always big hype and, of course, we also have to be really critical. People push for good news, we consume stuff on social media, LinkedIn, and it's hardly questioned. If you go on LinkedIn – I try not to be too much on LinkedIn, because you always get the impression everybody has solved everything and we’re basically the last ones. Then you go to clients, and you say, “Oh, hey. I've basically traveled the past for 10 years.” Because I have to make some effort to explain things, bring in new concepts, and get new ideas. So it's almost like a hot and cold bath.

But of course, there are many people who talk about of MLOps and actually have no clue what it is. Sometimes, if you look at larger consultancies, like really big ones – I have met many people or former clients and look at what they delivered, I always see, “Okay. Yeah. You basically have no idea. You basically just copy-pasted and did something. You don't have an understanding.” When we discuss MLOps, I don't put big letters there. I just put “There was this great paper on MLOps from the KIT with the whole detailed process.” And then we just go there. “Okay. Where are you now? What can you already do?” For example, data exploration – everybody can do that, at least if you considered MLOps. You already probably have part of a research plan and then you can see, “Okay, what do you already have to solve? Which pieces of the puzzle are already in place? What do we have to work on?”

We do a lot on finance or insurance and, of course, there are also regulatory measures. So, okay “Do regulatory measurements also fit in?” Of course, if you have this nice, detailed process you see, “Okay, before we release a new model, probably somebody from the department has to sign off.” Not because we don't trust the tech, but it's part of the regulatory process. Then you can just see, “Okay, this makes it a whole picture and we already discussed how we can solve it, what there is, what’s not there.” Because hype – nothing is a self-thing for a purpose in itself. MLOps – I love MLOps, because it just saves a lot of time and helps build better things. So I think it's the right way to go, although it's a lot of work for most companies.

# Thinking about the mid- and long-term when considering solutions

58:26

Alexey

**MLOps – would you say it's more about processes? It’s about the processes that you have to follow rather than tools. The impression I get from all these companies that offer MLOps solutions is that they generate a lot of talk like “Okay, *we* are the MLOps solution.” But it's not just about tools, but rather how you structure your processes. Right?**

58:51

Alexander

The problem is also that they always want to say, “Okay, you need to use the platform. You need to put the data there.” And there are many companies. We can't… we don't trust startups. Of course, you maybe have the funding and maybe you're gone next year and you have the data. Also we plan for mid- and long-term. We're not looking for, “Okay, we need a solution that works short term.” Of course, everybody's happy if there's a quick solution for everything. But we need to be able to run long term, because if you have millions of customers, you cannot say, “Oh, sorry. Wrong startup. We can no longer deliver this stack.” Or “We have to do it manually now.”

There are many things to consider and, of course, many of these platforms are also limited. As I mentioned in the beginning, freedom is very important to most clients – to say, “Okay, we need this feature. We want to implement it.” Begging the software vendors for that and waiting for responses is no way for us. Everybody has the experience where you file tickets and you never get an answer, even if it's something urgent.

This is basically the thing that I also mention because I would suggest MLOps only to companies if you have a decent team of experts around the company. I would not suggest “Oh we want to move into data science. Now we have hired three junior data scientists and we should do everything probably from the very beginning and do MLOps.” I always say this is a very bad idea because MLOps is complicated. It's a lot of work to get everything right – to get to this automation level. So you need to have in-house skills. Of course we can, at Königsweg, always help build the skills, help build MLOps, and other things – all in data and AI. But it's not an easy problem. Of course, it just looks easy because it makes sense. But there are a lot of details to work out.

# Finding Alexander online

61:07

Alexey

**So if somebody has questions and they want to ask you, or maybe they want to apply to that position at PyData that you mentioned, or maybe they want to ask for your advice to help their companies – how do they find you?**

61:23

Alexander

You can find me on LinkedIn. I’m the only Alexander Hendorf. Alexander Hendorf is actually really easy to find. You can find me on LinkedIn, also on Twitter – although I very often miss messages from Twitter. Probably the best way is just to at me, or write to me on LinkedIn. Or just, if you're at a conference, just come by and say hello.

61:47

Alexey

**At any PyData conference, right? You will be there.**

61:52

Alexander

No, I think not.

61:53

Alexey

**Just European ones, right? [chuckles]**

61:56

Alexander

I'm actually pretty close to PyData in Miami, but we are on holiday. Because that's on the 27th of September and actually I'm close by, so I was tempted, “Oh, I could go there.” But actually, we have other plans. We actually have plans for Disney World, so I cannot say I will go to PyData in Miami. [chuckles] So yeah – mostly the European ones. I’m sure I’ll also be at other ones around the world. Again, we're still coming back from COVID and pandemics.

62:33

Alexey

**So what’s the next one?**

62:34

Alexander

The next one is EuroSciPy at the end of the month. I’ll be there. It's also close by. It's only two hours by train.

62:46

Alexey

**Thanks a lot for the chat. I see that we went a bit over time. So yeah, thanks for joining us today. It was fun – a really nice conversation. Thanks a lot. Thanks for sharing your knowledge, your expertise. And thanks, everyone, for joining us today as well and for asking questions, for listening.**

63:03

Alexander

Thanks for organizing the podcast.

63:06

Alexey

**Yeah, well –online is not as difficult as offline. I cannot imagine what you need to go through to actually organize things offline. Because online it’s just a Zoom call and that’s it. But offline, that's an entirely different level. So I really admire your work. Thanks for doing this.**

63:23

Alexander

[chuckles] Thank you.

63:25

Alexey

**Okay, well. Have a great weekend, everyone. See you soon.**

63:27

Alexander

Thank you so much.