# Audience Poll

0:53

Alexey

**You might notice that there is a poll right now in the live chat, which asks “What’s your role?” Please take a moment to answer this question.**

1:26

Alexey

**We have 21 votes. Engineer Scientist – Individual Contributor is 85%. Not working yet is 9%. Manager is 4%. And Director is – how many do you think?**

1:41

Andreyy

1%? [laughs]

1:44

Alexey

**No. It’s 0. [chuckles]**

1:46

Andreyy

Zero. All right, yeah. I guess this will be very, very helpful. Every time we speak and do our sessions with my team, even with the individual contributors, I always like to give an outlook of what their jobs could look like as they progress, as they grow – how it's going to be next and really understand how organizations work. I think that could be helpful for everybody, even in an individual contributor role.

2:29

Alexey

**This week, we'll talk about building data science practice. We have a special guest today, Andrey. Andrey is a Director of Engineering at Honeywell, where he leads the advanced technology group within the safety and sensing business. He's working on developing technologies and solutions based on signal processing, perception, computer vision, AI, machine learning for healthcare, industrial, logistical market verticals. You're doing quite a lot of work. You've been with Honeywell for five years, right? Andrey also lives with his wife and son in Dallas, Texas. It's a pleasure to have you here, Andrey.**

3:10

Andrey

Thanks, Alexey. It's a pleasure to be here. I hope we will have a great session here.

# Andrey’s background

3:16

Alexey

**Yeah, we definitely will. Before we go into our main topic of building data science practice, let's start with your background. Can you tell us about your career journey so far?**

3:26

Andrey

Sure. Like you said, I'm a Director of Engineering at Honeywell. I lead and support what's called the Advanced Technology Group, which is a sort of a composition of several teams. We are working on novel technologies that we later introduce into our product line. So we work with a lot of signal processing, like you said, perception, vision, etc. Across the last five years I worked across and supported multiple organizations. I came to lead our AI adoption initiative and later transferred to build our robotics group, where I led our perception team for a year. So it's been a good journey of work through multiple advanced teams and building maturity and building practice into those teams. In terms of my career, I've been coding since I remember myself probably – at seven years old.

I've always been kind of an engineer in my heart. But my formal background is actually not in computer science. My formal background is organizational development management in the context of IT and technology organizations. I have a scholarship and multiple degrees in organization development and management. Overall, I started as a software engineer, and I saved my paycheck on the side to fund a few businesses. Some of them that I successfully failed. [chuckles] And some of them were later acquired. During one of those businesses, where I worked as a technical co-founder, that business was around scanning and digitizing retailer catalogs for coupons. It was back in Europe. We were essentially scanning those coupon printed paper catalogs and we were extracting the information from those catalogs. We were offering that on our website and that required a lot of, first of all, image analysis and also recommendation models. That was in 2013-14.

This is where I fell in love with data science. This is where I really started my journey with data science – with conventional data science – data analytics, and then later into neural networks, of course. It was interesting, because, after I had my own set of businesses, I went to another extreme – I went to Honeywell. And Honeywell in Europe and the Middle East, it might not be as well known as in the US. Honeywell is a Fortune 100 company with a 100 year old history and heritage. Before it was well known mostly for industrial hardware – industrial equipment. Some of you might ask, “Well, what brought you from doing your own startups into a multinational conglomerate with 100,000 people?” I joined Honeywell five years ago, when the company got its new CEO, who put the company on a trajectory of digital transformation. Really what it meant is moving the company from being hardware-oriented into more software industrial-oriented.

Essentially, it was changing our model from selling one-time hardware to selling services and subscriptions that are enabled by that hardware. That meant, really, sensorizing and connecting all of that equipment, pushing data into the cloud, and doing a lot of analytics, doing a lot of signal processing and doing a lot of intelligence in the cloud, that allows you to actually multiply the value of those solutions. And that really sold me on that. I realized that there would be a lot of scope of doing analytics, doing AI, and we could be really at the beginning of it. So that’s really what sold me on my role.

Everything that we work on here is really high impact. You are, in your everyday job, impacting millions of customers. Over my last five years, I've worked with logistics warehouse robotics, I’ve worked with smart cameras, or some cool technologies for sensing. Most of what we do is vision and perception, but we also worked into other modalities like natural language and voice and predictive analytics overall.

8:54

Alexey

**I'm just wondering, what kind of industrial equipment do you work with? Is it like conveyor belts or things like that?**

9:02

Andrey

Yeah, this is one of the product lines. Honeywell is a really distributed organization. It's actually like a Russian nesting doll where you have five big business groups at the top – aerospace, smart connection, performance materials, safety products, and our own cloud platform. So it's really five big companies within the umbrella of one company and then each of those companies split into their own businesses and units. It's a very complex structure of businesses and portfolios that you need to navigate.

9:42

Alexey

**That's a very interesting name. I would assume that they do like breakfast cereals, because the name sounds tasty. [chuckles]**

9:56

Andrey

No. [chuckles] [unintelligible] In the US, the company is well known. Almost every building has some sort of Honeywell product – every commercial building. I bought a house last year, and I found a gas meter from Honeywell and a Honeywell thermostat. So, when I try to sell Honeywell to the engineers and managers that I hire, I usually talk about that, which is true, that essentially, anywhere you go – any building, airplane, car, even space shuttles, you would find something from Honeywell.

10:39

Alexey

**And you develop some things for these gas meters and things on airplanes? Or only for these assembly lines, right? You're working on that. [cross-talk]**

10:51

Andrey

I work in the business group, which is called Safety and Productivity Solutions. There are things like, again, smart cameras, smart sensors, robotics for warehouses. You go to Starbucks, or you go to Ikea, you would see that the barcode scanner at the POS terminal would be built from Honeywell. In your daily life, you won’t notice it, but really, you touch Honeywell products on a daily basis.

# What data science practice is

11:22

Alexey

**Quite interesting. I know nothing about this company up until now. You are doing quite a lot of interesting stuff. So the topic today is building data science practice. But I was wondering, what does “data science practice” actually mean? Is it a data science team, or is there more to that?**

11:48

Andrey

When I think about practice, I think about the widespread adoption of best data science and machine learning engineering practices across the organization. It might not seem like something special in the context of small organizations, where you have just one product team, but when you think about multi-product team organizations – organizations that are distributed – it's not a straightforward thing to make data science work efficiently.

Really, it's about breaking organizational barriers, not allowing for silence – if you want your data science strategy to run efficiently, especially in the very beginning, you cannot allow teams to use multiple frameworks, or use multiple tools, or use multiple clouds. Really, that makes things inefficient. So this is what I think of when I talk about data science practices – it's aligning everybody on the same page. Thus, teams not only work efficiently within the team context, but also within the organizational context.

It's just about maturing up the organization and to shift from ad hoc pilots, where teams constantly are building pilots, pilots, pilots, but they don't push those pilots into production. And “data science practice” is about understanding how to shift from the constant pilot stage to a mature stage, when you ship and you productize.

# Best DS practice in a traditional company vs IT-centric companies

13:46

Alexey

**I’m wondering, in your particular case at Honeywell – Honeywell, strikes me as a more of a traditional company, where IT comes second (AI comes second) while there are internet companies, where it's much, much easier to get data and start using machine learning. For you, I guess that was an extra layer of challenges, right?**

14:15

Andrey

Exactly, exactly. When I was talking about starting five years ago, the company took a turn of saying, “You're no longer going to be focused just on the hardware.” With hardware you can think of – you build the whole hardware, let's say, a gas meter. You sell it one time. Then you might support it, you might maintain it. There's rarely a software component. But this was like five, six years ago, when the company turned and said, “We're going to be doing a lot more software now.”

Essentially, this is where the complexity of integrating data workflows and algorithms that are based on data come into place. Really, it was such an honor of being a part of putting the company on the trajectory of change. First of all, I won't say that everything went perfectly well. We learn from our mistakes. But it's also always a very difficult thing – to make a shift like that in a company with a 100-year-old history and 100,000 people. It's a humongous effort. And this is where you make sure that you learn… There's so much complexity here that you learn really fast.

15:42

Alexey

**I'm really curious now that you mentioned it. This is not what I was going to ask initially, but now that you mentioned a gas meter – you just sell it and support it and there is no way you can include software. So can you actually include software into a gas meter?**

16:03

Andrey

Right. You have, of course, a long tail – you have products that are more conventional and less intelligent, but you start building intelligence into them. Well, a better example would be – we started working on some advanced sensors for the quality of the air, for example. Before I worked on this problem, I didn't think that that was such a problem. Well, you're breathing air, right? How could that be dangerous? But then, when we started actually looking under the microscope, “What are the different things that we are breathing in? Mold and pollen and pet dander and dust.” You get really scared about what you're really breathing in or things.

When you buy a house, you realize some things as well. I have a cat allergy and when I moved in, I realized that there was a cat living here. We had to do a deep clean there. So, for example, we are building a sensor that, with the push of a button, can tell you all the micro allergens that are in the air. So what you're actually getting from making those things connected is, first of all, you now can process the data in the cloud and you can run massive, massive signal processing and analysis on the data that you process. And you're not only able to run the workloads in the cloud, but you can actually get this data and fine tune your model, of course. That allows you to really build algorithms that you wouldn't be able to, literally. Otherwise you would need a whole computer with an industrial GPU on it and this device would cost several thousands of dollars and things like that.

18:06

Alexey

**You cannot put it physically in a gas meter, right?**

18:11

Andrey

Exactly, exactly.

# Getting started with building data science practice (finding out who you report to)

18:13

Alexey

**Coming back to data science practice – you mentioned that data science practice is a set of data science and machine learning practices, or how you spread the adoption of these practices. Then you mentioned not having silos, having the same tools and processes in different teams and things like that.**

**It sounds like a very good thing to have, and I'm wondering – say I’m joining a company as a first data scientist (data science manager, or maybe director, or just as a data scientist) and we want to build it. So how do we build these data science practices? What are the steps to get there?**

19:06

Andrey

Right. Here, of course, I might say “it depends”. But it really depends on multiple factors. I won't say there's just one path. I sort of have a mental model of going from low maturity to high maturity, or like I say, “You crawl, you walk, and then you run.” A lot depends on the size of the organization, of course. A lot depends on what would be the focus – where are you going to deliver the value? Actually, the first major thing when you're interviewing with a company – or you're already working in a company – it's good to reflect on who in the organizational hierarchy the data science team is reporting to. I think I never actually asked myself questions like this when I was just an individual contributor, but it's hugely important.

It's important because, depending on which executive the data function rolls into, this data group could actually do very different things. The example would be – if the data function rolls into the CTO (Chief Technical Officer) that would most probably mean that the data group will be focused on enhancing the product capabilities. CTO is responsible for building an offering, or building the product. So if you are reporting to the CTO, that means that your function goals would be to enhance the capability of the actual product that you’re working on.

But this is not the only thing. Because you could also, for example, report to the CIO (Chief Information Officer). In this case, very likely, you would be working a lot on the internal optimization workflows, like the optimization of efficiencies of your internal ecosystem, instead of working on the product. Or the data team could report into the Chief Marketing Officer. Then it would mean that it's mostly an analytics function around marketing and sales, and customer interaction. So that's an interesting question to reflect on. Because, really, whoever you report into, that means that you will be helping to achieve the goals of that particular executive.

It's probably step zero to really reflect on this and say, “Okay. Where can the data provide the biggest value?” and that does not necessarily need to be either/or. There are organizations that report into the CEO or. Then in this case, if you're reporting to the CEO, most likely, it would mean that you will work across all of those functions. The data will have their own place both in product engineering, both in internal optimization, both with sales and marketing and customer interaction.

This is a very important question to reflect on because that will really define what kind of profile of talent you will be working with, and what the kind of profile of algorithms that you will be working with. The example is – we work on product, and the modality of the algorithms that we work on is mostly signal processing and computer vision. This is because we work a lot with the sensing aspect of our work. But then, we also have a team in Honeywell that works on the optimization of internal processes, like demand prediction, for example. And that team doesn't do any computer vision. That team only works with, essentially, fairly conventional predictive analytics algorithms. There is nothing wrong with that, but it's just a different types of algorithms that you will be working on. So it's good to actually understand that, because that answer will align you with what kind of talent, what kind of algorithms, what kind of technology, and the infrastructure that you will have to build in order to fulfill those goals. I hope it makes sense.

23:43

Alexey

**Yeah, it does. I'm just wondering. The company where I work, called OLX, which is an online marketplace – when I joined the company, the data function (my manager/team) was reporting to a CPO (Chief Product Officer). In this case, I think it's a similar idea as reporting to a CTO, so improving the product. But here, in this case, “product” has its own vertical. I guess in cases where you report directly to a CEO, it makes you a Chief Data Officer. Right?**

24:26

Andrey

Now, some organizations call it formally a Chief Data Officer, or it could be VP of Data. But reporting to a CEO means that you will not only have a focus on engineering, or you will not only have a focus on internal IT and things like that. Your function will go much, much broader. It's really about “Who is going to be your executive sponsor?” Your executive sponsor is going to lead and assign particular goals and particular initiatives for you. So that's hugely important. But as we go through this, you asked about the steps. My mental model is that you go through low maturity, medium maturity, and then high maturity.

Low maturity would mean that you come into the team and you start building a team. Usually, it depends on who drives that. There are situations when the executives drive that, and they will hire senior roles, and the initiative of developing this practice would come from the executive level. But many times the initiative of building data into products, or internal capabilities, would come from the engineers themselves. I've been there myself.

When I came to Honeywell, we had sort of a similar situation. Some of the initiatives that we were building really came from the software engineers, who, at some point of time, got inspired by data and they started working with machine learning algorithms or other conventional data algorithms. Another approach would be engineering managers would think of additional capabilities and they would start hiring data scientists, thinking that they can just hire a data scientist, plug them into the team, and then overnight – magic will happen. Overnight, the products will get some magic intelligence, artificial intelligence and that type of… [cross-talk]

27:01

Alexey

**A lot of profit.**

27:02

Andrey

Yes, yes. But it doesn't happen this way. Data science or artificial intelligence are very multi- or cross-functional types of stacks. This is one of the misconceptions – where you can just hire a PhD, and overnight, you'll get the benefit. I think many organizations underestimate the complexity of it. It really impacts the talent that you have. There's more than just data scientists – and we can talk about this – there are more research scientists, there are machine learning engineers, there are data engineers.

There is impact into the infrastructure, there is impact into how you run processes, there is an impact in the whole mindset shift, where going from a sort of pre-planned “waterfall” type of approach to engineering – into more iterative engineering. So those are things that many people, sometimes from the conventional software background, don't really know about. This is why there are often just overly high expectations and this is why things take so much longer than originally assumed.

# Who the initiative comes from

28:19

Alexey

**In your opinion, what happens more often – that it comes from the engineers (from the bottom up approach) or is it more like an executive listens to some consultants from McKinsey and decides to build this practice? What happens more often?**

28:41

Andrey

Yeah. Being data-driven, I don't have those stats.[chuckles] But I think it's… [cross-talk]

28:48

Alexey

**Anecdotally.**

28:49

Andrey

Right. I think it's a combination of that. I see both of those situations. The same way, like you're talking about, both McKinsey consultants and from engineers. But CEOs and CTOs – we're not talking about something that came about just yesterday. Data science got popularized – I think the peak was probably around four, five years ago, when it was really at the peak. Right now, almost everybody knows about it. But also, I see a lot of engineers – software engineers with conventional backgrounds – because AI has become such a hot topic and there is so much democratization and the tools that we have right now. Like the example with stable diffusion, when you can just download a Google Colab notebook, run a few cells, and you will have the whole thing in front of you. I think many people just really feel very empowered by that, get excited, and sometimes even overexcited about it.

This is something that I really often try to watch out for, both for myself and for my team. Because if you get overexcited about something, you sometimes get really biased, and instead of really working through what's important for the customer, you start working on something *just* because you want to try this technology. Then AI becomes not a solution for some customer problem, but it becomes a shiny thing that nobody knows what to do with. When you think about integrating a data science workflow and building up data science capabilities, you really need to start from “How can I really improve my customers’ experience?” Instead of, “Hey, I have this shiny technology! Let's find where we can plug this technology into!” [chuckles] I think so many people just get super biased about this.

# Finding out what kind of problems you will be solving (Centralized approach)

31:02

Alexey

**Speaking of stable diffusion. This is indeed super cool. The other day, I played with this and generated a dataset with dragons and dinosaurs. To me, they always looked the same and I thought “Okay, maybe we can just get some data with dinosaurs and dragons.” It's still… The pictures are still creepy. But sometimes they're really good.**

**Coming back to this building practice – let's say, we join a company and we figure out which executive we report to, with the options that you mentioned being CTO, CMO, CIO, CEO – all of them. So what happens next? What do we do next? We figured out that, let's say, it's a CTO. What's next?**

32:00

Andrey

The way it *usually* works is that – the right thing, of course, is to understand what kind of problems we want to solve with data or what kind of capabilities we want to build with that. But it doesn't always work like this, as I said. There are some grassroots efforts (bottom up) when engineers start coming up with new ideas right there and start building some proof of concepts. Usually, the first stage is what I call the “crawl stage”. Before you walk, you’ve got to crawl. This is where teams start to build those proofs of concepts. Usually, it's with a company that still has not built a mature data practice. This is a really big challenge – to move some of those proofs of concept into production. Because, again, the expectation is “Hey, let's just hire PhD data scientists. They will just build a model. Boom, boom, boom. There we go. We will have this new magic functionality.” But again, it doesn't work like this.

This is exactly what the situation was when I came to Honeywell five years ago. We had some data scientists here and there, who were hired by conventional software engineering managers who didn't know anything about data. So the expectation was that we were just going to hire them, plug them in, and they will start working. But it's more than that. First of all, it's really about the fact that it's very easy to build flashy proof of concept demos, but it's very difficult to get them into production. First of all, building a model for the flashy demo is done on a small dataset. You can probably get some data points that you can build a model with. But to make it a really robust model, you’ve got to have a different level of volume for your dataset. That's always a problem, “Where are we going to get the data?”

Unfortunately, companies that are not as mature don't prioritize collecting data early enough, which has to be really prioritized right there when you actually build a project passport. Step zero when you're thinking about the project – when are you thinking about a feature or function that you're going to integrate into an existing project – you’ve already got to start thinking, “Where are we going to get the data on top of which we are going to build our model?” Companies don't prioritize that. To build an initial model, it takes some time, but it takes 10x more time to actually get it into production.

There are many other reasons that really become problems. Things like the engineering component of building data products are different from conventional software engineering. Just by building the model, you cannot enable the feature. You need to integrate this model, you need to deploy this model, you need to watch and operationalize this model – you need to be able to have the closed loop of retraining this model based on the new data that is coming in. Usually, this is not how conventional software engineering works. So you really need to change processes, change the infrastructure and that takes time. This is why, often, you have a hard time really deploying those demos.

I've been wrong many times on this myself, and this is exactly how I started myself. But you reflect on all of those issues and you start fixing them one by one. In this “crawl” stage – in the beginning stage – I'm not a big fan of having many different projects and seeing what actually works out. Because you have multiple teams, and when you have multiple teams (and multiple scientists and engineers) and it's more difficult to manage them, my approach is having as few projects as possible, on which you can build the whole end-to-end process.

In this crawl stage, the idea is usually to just get one project out the door end-to-end. One project that would become sort of like a poster child – an example – for the whole full cycle of collecting data, running experiments, identifying which model is the best, changing the infrastructure, and having the roles of the people who would work on the infrastructure to be able to product productionize those models, monitor those models, retrain those models, and essentially having the whole end-to-end cycle.

This is where I would say data teams should put the major focus on – having at least one project that is successful, instead of having 10 projects that fail. Because if you have 10 projects that fail, what actually also happens is that data science gets a bad reputation. People get a lot more negative about building data-enabled products. So, here, I would put most of my eggs in one basket and run it end-to-end. This is where you switch into the next phase of data maturity and you build a centralized practice. That means you want to now get all the efforts and all the scientists in one organization and really start building, when working on further process… [cross-talk]

38:23

Alexey

**You mean a team. Right?**

38:26

Andrey

Right, right. In early stages, it's good to collect all of them into one team, or all into one organization, and centralize them. What I mean by that is that you need to build a set of best practices. When all of those people are under one team, it's easier to popularize those best practices and integrate and deploy those best practices. What do I mean by “best practices”? It starts with talent acquisition and hiring. It’s about understanding, “What are the actual roles? What are the differences between research scientist, machine learning engineer, data engineer, data quality engineer, and other professions?” It’s also about actually having job descriptions for those. Because when you start, you come to HR and they don't know who they need to hire. When you hire the wrong people, everything is wrong after that.

It also starts from aligning on a particular set of infrastructure and tools. “Which cloud are we going to be using? Are we going to be using cloud? Are we going to be using on-prem? Are we going to be using our own deployments? What kind of tools are we going to be using? Are we going to write our own experiment tracking system? Or are we going to be using other tools on the markets from Weights & Biases, Neptune, Comet, (and so on and so forth)?” How are we going to be productionizing and deploying those models?” and things like that. When you work within one team, it's a lot easier and faster to build those best practices and really align the team on those best practices.

The trick is this centralized approach. This is how I think – it's really subjective. It's not the best way of managing data organizations, but it's the step that you have to go through on the way to higher maturity. In a way, you start with the ad hoc approach, where you build one major proof of concept, you go to a centralized model, where you get the team together and you build a set of practices by working on more projects. But this is not the most efficient way of running data teams, in most cases. Just because, when you have a centralized team, and let's say, a product manager comes to you and says, “Hey, we need to build this (let's say) recommendation engine for our website. Can you give us a data scientist?” So you have a list of projects that you’re working on as a centralized team and then you need to market your team internally, so product managers or engineering managers come to you, and you act sort of like a resource pool.

But then it doesn't work most efficiently because there is a queue, essentially. You need to prioritize those projects. And how do you prioritize those projects? But also, you get a data scientist or machine learning engineer, or data engineer – today they’re working on one project, and tomorrow, they’re working on a different project. So they don't really work consistently with one team, they don’t necessarily know how people, or how those teams, operate and act. They don't establish trust and respect with those teams.

What I also found out, through a lot of mistakes and hard experience, is that when you have a centralized team, the performance of your scientists is treated by their own scope. It's gonna get clearer now. If you actually push that scientist right there into the team, and have that scientist actually report into that team, and be formally a part of that team, the performance of that scientist is going to be treated more based on the performance of the whole project.

On one side, you have a centralized team, and the performance of the scientist is rated on the performance of their work. But if they work in a particular team and report to a particular engineering manager, for example, their performance is going to be treated more as the performance of the overall project, which is something you actually want. You don't want your engineers and scientists only to say, “Hey, this is my type of work. This is the only thing that I'm responsible for. I don't care about anything else.” You actually want them to be the drivers and the fans of the project itself. You want them to think the project’s success fully relies on them. This is where the third modal equation comes in…

43:39

Alexey

**That's a lot of information. I want to try to summarize everything that is said before we move on to what you call the “run” phase.**

43:49

Andrey

Decentralized approach.

43:51

Alexey

**Decentralized, yeah. First, we need to figure out who is our executive sponsor. This also helps us to understand which problems to solve with data, what kind of things we will work on. Then we can start with a proof of concept – not *many* proof of concepts. It's better to select as few as possible – let's say a single one – but we need to make sure that it is successful at the end. We don't want to fail because if we fail, then data science gets a bad reputation and then we might lose the trust of our executive sponsor who was rooting for us. We don't want to do that because the executive sponsor really believed in us. Therefore, we need to try to meet their expectations.**

**We do this POC, which often comes from engineers who want to do something cool, and then they pitch it to their managers, and then the project starts. Eventually we have a finished POC and then it goes to the next phase, which is the centralized approach. We don't have any data and we need to build all these pipelines, we need to productionize the model – and we do this centrally, because we want to make sure that we build this set of practices that you mentioned. Then it's important to understand what kind of roles we need, what kind of tools we want to use, and all these things that you just talked about.**

**However, the centralized approach creates problems, because you act as a resource pool, and sometimes there are not enough resources in the pool and all these other problems that you mentioned. This brings us to the next phase – the embedded teams, or decentralized teams, where a data scientist or a data person sits in the team. This is where I stopped you.**

# Moving to a semi-decentralized approach

46:04

Andrey

When you say “where the data person sits,” the important part is who they report into. They can still work on those projects. But the question is, “Who do they report to?” Do they report to, let's say, the organization of a chief data officer, who is central? Or do they report to the engineering manager of that particular product? And it's a big difference, because your manager is the one who rates your performance. In this case, it's usually better that you actually move from a centralized model by taking all those people who used to be who used to be in your centralized model – in your own team – and then you say, “You're now going to report to this engineering manager and you're going to be working for that product (or that function).”

It's not that it might sound like something difficult or heartbreaking. You used to report to one manager and now you report to a different manager. But really, in my experience, many times it actually came very naturally. Because as people work with different teams, they create some affinity and some experience towards particular problems, or particular teams, or particular business domains. And when you transition from centralized to decentralized, it's actually a much easier approach because many times people will say, “Yes, I want to work for that product right now.” So that doesn't create as many friction points.

This is actually what we went through – we used to have one centralized team and then we said, for example, “You used to work on this particular set of problems for this particular business, for this particular product line. You are now going to report to that engineering manager and you're going to be working for them.” You essentially spread your team across and you have them work on a particular project. [cross-talk]

48:08

Alexey

**And what do you do? Do you just kick back and don't do anything? [chuckles]**

48:13

Andrey

Oh, no, no, no. This is why I say that I've never seen fully decentralized models work ideally. Usually, it’s what I call semi-decentralized. You still have to leave some space. You still need to have one central function that will be responsible for some of the functions. For example, the hiring, or the infrastructure – some of those things still have to be still. But the engineering, for example – engineering teams and product engineering teams – it's better that they work with their engineering managers and their product teams.

There is still some level of centralization. Sometimes this model is called “hub and spokes,” meaning that there is one hub – you still have a chief data officer that's responsible for some of the high level functions and standards, but then you have different teams across organizations that work almost autonomously. You still build bridges between them, you still kind of work with them to understand what kind of problems they have, what kind of roadmaps they’re working on. But what's really important, I think, is that you have to have the right set of steps.

You have to go *first* to centralized and then decentralized. The reason for that is, when you’re centralized, you need to build that set of practices and then those people take those practices to their own teams. They don't start from zero when they start their product teams. This is hugely important.

49:54

Alexey

**And your role, I guess, is making sure that they have this common set of practices, and now that they’re spread across the organization. They might come up with their own thing, which is fine. But you still want to maintain the same set of tools, the same set of practices. It’s not like… [cross-talk]**

50:14

Andrey

There are still some things. And it's not always the same. It doesn't mean that everybody needs to use PyTorch. Some will use Python and some will use Tensor. But I'll give you a good example. Vendor relationships. When you work with tools and the procurement of tools, the central function is much more efficient in identifying which tools we want to procure for those teams to use, rather than having those teams actually decide what tools they’ll use. Because you don't want to have different vendors for the same things in different groups, for example.

Again, it's not that black and white, but this is how it usually works more efficiently. As a central function, knowing what kind of problems your teams are dealing with, you can say, “Hey, how about we use *this* particular experiment tracking or MLOps platform.” And then, because this is the way you can actually get the best deal, and you then can help those teams consume those products. This is one of the examples that work very well. Or with data annotation or image annotation – we do a lot of image annotation. What vendor are we going to be using for data annotation? Things like that. There are still some functions in the central org that are more efficient to be solved by them.

# Resources to learn about data science practice

51:47

Alexey

**We have quite a bunch of questions and not so much time. So we should probably go to the questions and try to answer them. The first question was, “Which book would you recommend for us to learn more about building a data science practice?” Are there books like that at all?**

52:03

Andrey

Can I post a few suggestions in a LinkedIn post to this event? [Alexey agrees] There are quite a few. I read a lot on this. I’ve got some books that I read some time ago that I wanted to recommend. They were fairly amazing. I have a hard time right now to quickly identify it.

52:33

Alexey

**Yeah, we can include the links in the description later on, when you find them.**

52:37

Andrey

I promise to get back to you.

# Pivoting from the role of a software engineer to data scientist

52:39

Alexey

**Okay. Another question from Adonis, “You mentioned that software engineers pivot into data science. How do they actually convince their manager to do this?”**

52:56

Andrey

It's a good question. If you're a software engineer and you want to pivot into data science. I actually have a long post on LinkedIn about this. But my advice, as always, about this is – it will be very hard for you to get hired as a data scientist from scratch, if your previous position is a software engineer. So my advice to you is try to convert into a data scientist at the same company you're currently working with. Thus, you actually get a formal data scientist title, and you work through and get a few years of experience in it, and *then* you can either stay in this company or switch and get hired into another company with a formal role of data scientist.

Now, to get transferred is really about finding the right problems to solve and really solving them. If you can come and offer – let's say, you're building a product recommendation algorithm and your team doesn't have any experience with it – and you can really, in your free time, dive into the subject, gather the data, build a prototype, and really show your team that you can build a mature model. It's not going to be easy and it will take some time. It's not that just because your software engineer, they will say no. Sometimes they will say no, but it's about being persistent in offering yourself as a problem-solver for data-heavy problems. Also, within your organization, move into teams that have more data. Think about it from the perspective of “Where do I have the data and what are the problems that I can solve with that data?” The teams that have an abundance of data, usually have problems that you can solve with this data.

55:07

Alexey

**Does that usually happen at the POC stage, the centralized stage, or decentralized stage? Does it even matter?**

55:15

Andrey

It doesn’t. The more mature the organization is, the harder it would be to switch, but it's not that you won't be able to switch. I have a few examples of people who were software engineers, who were actually just persistent and helped solve hard problems using data. Then at some point of time, you come to your manager and say, “Hey, I've essentially been a data scientist for the last two years. Can I work with you to formally change the title?” We had use cases like this.

55:51

Alexey

**We actually also had use cases like that in the company where I work. A person was a data engineer and then he said, “Okay, look. I’ve actually been doing ML engineering for the last two years. Can we make the switch formally?” [Andrey agrees] I guess it’s easiest to do at the POC stage, right? You just say “Okay, there’s this cool thing. Let’s try it.” And then this thing works. All of a sudden, one year later, you're just doing this thing.**

56:19

Andrey

Right. But if you're a software engineer, you will probably (most likely) underestimate the complexity of it. [Alexey agrees] You might just build, again, a model that works for the demo. But really integrating it in a robust way – there is a long way to go from a demo to a robust model. And you will probably underestimate that. It's not weeks and it's not months from that level.

# The most impactful realization from data science practice

56:44

Alexey

**Another question we have is, “What is your favorite data science practice so far?” I guess we've talked about all these best practices. What do you think is the most important one to have?**

57:02

Andrey

Hm. Interesting question. I think what's most impactful, and where many people make mistakes, is really that – if you hire a PhD scientist, you should not rely on what they make to productionize. You should not rely that the code that they write in a Jupyter Notebook could be transferred into production. For this, you will need another stage of machine learning engineer and/or applied data scientist, who would then recrunch all of that code and put it into production. I think many organizations, many people (many managers) with conventional backgrounds don't understand that. There are actually two extremes of developing a model and having that model run in production. And you have to put other roles in place to ensure that you can make this code robust to productionize it.

58:17

Alexey

**That’s probably not the best practice, but more like a realization and a way to set expectations – “Look, if we hire this person who has a PhD in machine learning, it doesn't mean that we’ll automatically get rich tomorrow.” So managing these expectations, and then saying that, “We actually need a *team* of engineers to make it work.” That would be your answer, right?**

58:45

Andrey

That essentially comes down to having clear roles – there is a research scientist and distinguishing that research scientist from a machine learning engineer. I know we are running out of time, but I can tell you one example. I used to work with PhD research scientists and every time I would come to them with a problem, they would say, “Let me go and read some white papers.” Many times you don't need to read white papers. There is *this* solution up front for you that you can do. But I see a heavy reliance on “Hey, I'm gonna go for two weeks and I'm gonna read white papers before I'll give you a solution.” And this is what I clearly think is a research scientist mindset, which does not work well when you productionize anything. It's great to have them, but you have to apply additional roles in order to transition to production.

# Advice for individual growth

59:44

Alexey

**Okay. Before we finish today, maybe you wanted to mention something but didn't have a chance?**

59:52

Andrey

Well, we talked about so much already. I hope it resonated. And I know we had a lot of scientists and engineers in the audience. Again, my advice would be – if you want to grow, you need to think from the perspective of “The higher you go in your seniority, the more impact you will have across the stack, across the team, across the business, across the organization.” So, my advice is, spend some amount of time learning what actually happens beyond the scope of your role. That will propel you faster to the next level.

Again, people are promoted, not because they will learn the skills they will require at the next level, they are promoted because they already exercise the skills that they would need to be at the next level. Be more forthlooking into what other teams and other roles are doing, and be more open about, “How do I make a broader scope type of impact beyond just what I'm currently working on?” And thanks for this chat. I hope that was interesting.

# Finding Andrey online

61:33

Alexey

**Am I right that the best way to reach out to you is LinkedIn?**

61:36

Andrey

LinkedIn would be best, yeah. I'm fairly active on LinkedIn, so please do connect.

61:42

Alexey

**Okay. Thanks for joining us today. Thanks for sharing all your expertise and experience with us. Also, thank you, everyone, for joining us, for asking questions, for being active. I see that there is a long discussion in live chat. Yeah, thanks for joining us today. Everyone. Have a great weekend.**