## DataOps and Christopher's background

**2:12  
Alexey**This week, we’re discussing DataOps again. Maybe it’s becoming a tradition to talk about DataOps once a year, though we missed last year. It’s been a while since we had Chris on the podcast. So, today we have a special guest, Christopher Bergh. Christopher is the Co-Founder, CEO, and Head Chef at DataKitchen, with over 25 years of experience—probably more now—in analytics and software engineering. He's a co-author of the "DataOps Cookbook" and the "DataOps Manifesto." It’s not the first time we've had him here. We interviewed him two years ago, also about DataOps. Today, we’ll catch up and see what’s changed in these two years. Welcome to the interview, Chris!

**3:18  
Christopher**Thank you for having me. I'm happy to be here, discussing all things related to DataOps, why it matters, and what's changed. Excited to dive in.

**3:31  
Alexey**Great. So, the questions for today’s interview were prepared by Johanna Bayer. Thanks, Johanna, for your help. Before we dive into DataOps, could you give us a brief overview of your career journey? For those who haven't listened to our previous podcast, tell us a bit about yourself. And for those who have, maybe a quick update on what's changed in the last two years.

**4:05  
Christopher**Sure. My name is Chris, and I'm an engineer at heart. I spent the first 15 years of my career working in software, building both AI and non-AI systems at places like NASA, MIT Lincoln Lab, some startups, and Microsoft. Around 2005, I got into data, thinking it would be easier and I'd be able to go home at five. I was wrong.

**4:43  
Alexey**You started your own company, right?

**4:47  
Christopher**Yes, and it didn't go as planned. The challenging part wasn’t doing the data work itself. We had talented people for that. The real challenge was the systems around the data. We had a lot of errors in production, and we couldn’t move fast enough to meet customer demands. I used to avoid checking my Blackberry on my way to work because I dreaded seeing problems. If there weren’t any issues, I’d walk in happily. If there were, I’d brace myself.

## Early challenges in data management and the pre-Hadoop era

**6:01  
Alexey**Was this during the Hadoop era, before all the big data technology boom?

**6:06  
Christopher**This was actually before Hadoop. We used SQL Server, and our team was so skilled that we turned SQL Server into a columnar database to make things faster. Even then, the core principles were the same. We dealt with databases, indexes, queries, etc. We just used racks of servers instead of the cloud. What I learned was that managing a data and analytics team is tough. I started thinking of it as running a factory, not for cars but for insights. How do you keep production quality high while making changes frequently?

## Evolution of DevOps and its influence on data practices

**8:29  
Alexey**Interesting. So, you mentioned DevOps. When did the concept of DevOps start gaining traction? How did it influence you?

**8:53  
Christopher**Well, the Agile Manifesto came out in 2001, and the first real DevOps practices started around 2009 with automated deployment at Twitter. The first DevOps meetup happened shortly after that. It's been about 15 years since DevOps really took off.

**9:38  
Alexey**I started my career in 2010, and I remember manually deploying Java applications via SFTP. It was nerve-wracking, just hoping nothing would break.

**10:03  
Christopher**Right? Was that in the documentation too? "Deploy and cross your fingers"?

**10:18  
Alexey**Almost, there was a page in the internal wiki on how to do that.

**10:29  
Christopher**Exactly. The question is, why didn't we automate deployments back then or have extensive regression tests? Nowadays, it's almost unthinkable not to use CI/CD or automated tests in software development. Yet, in data and analytics, that hasn't always been the case.

## What is DataOps and its importance in the industry

**11:53  
Alexey**Let's step back and summarize what DataOps is. Then we can talk about what's changed in the last two years.

**12:03  
Christopher**Sure. DataOps starts with acknowledging some hard truths about data and analytics: we're often not successful, and many people in these roles are unhappy. We did a survey with 700 data engineers, and 78% wanted their job to come with a therapist. Fifty percent were considering leaving the field altogether. Teams often fall into two categories: heroic, working non-stop but burning out, or bogged down in so much process that everything moves at a snail's pace, leading to frustration.

**13:22  
Alexey**So, the only option is to quit and start something else, right?

**13:27  
Christopher**Unfortunately, yes. When a team relies on heroes or strict processes, you end up with a few people holding all the knowledge. If they leave, the team struggles, creating a bottleneck. DataOps is about finding a balance. You don't have to live in constant fear of making mistakes or being a hero 24/7. There's a middle ground where productivity thrives.

**14:17  
Alexey**Fear is when you're scared of deploying changes because things might break, right?

**14:43  
Christopher**Exactly. Fearful teams often have excessive checklists and reviews. Heroic teams will deploy changes and hope for the best, ready to fix issues at any time, even if it's their kid's birthday. That’s not sustainable. As a manager, I’ve learned to praise the heroism publicly but privately work to ensure those situations don't happen again.

**15:52  
Alexey**So, DataOps involves processes and tools to help move without fear and avoid heroism, right?

**16:10  
Christopher**Yes. DataOps aims to reduce errors in production, whether they're caused by bad data, code issues, server failures, or delays. Automation, testing, monitoring, and observability are all part of this. By focusing on reducing errors and improving cycle time, we can eliminate waste and increase productivity. Gartner reported that teams using DataOps tools and practices are ten times more productive, which aligns with what I’ve seen.

## The current state of DataOps and its evolution over the past two years

**18:46  
Alexey**Two years ago, there was a lot of hype around MLOps. It brought attention to other areas like DataOps. Now, the focus has shifted to AI and LLMs, and it seems like DataOps isn’t talked about as much. What’s been happening in DataOps over the last two years?

**20:24  
Christopher**Good question. I think it’s important to differentiate between buzzwords and core principles. DataOps, much like DevOps, is built on lean manufacturing principles from the Toyota Production System. These concepts are decades old but still relevant. The marketing around new terms like Data Mesh or Data Observability often distorts their meanings, which can be frustrating. At its core, DataOps is about agility and system thinking—whether you’re working with data, ML models, or LLMs, the principles remain the same.

## Systems thinking in DataOps and managing day one vs. day two and day three

**23:56  
Alexey**You mentioned "thinking in systems." What does that mean?

**24:24  
Christopher**It’s about considering not just the initial build of a project but also how it will operate on day two and beyond. Day one is building something for the customer. Day two is running that system with new data. Day three is making changes based on evolving customer needs. A lot of data teams focus on day one, but managing day two and day three requires systems thinking. You need to build processes around quality checks, monitoring, and quick, safe deployments.

**26:13  
Alexey**Let's take a data scientist as an example. They pull data, do some transformations, and build a model. Day one is about getting that initial version ready. What happens on day two?

**26:54  
Christopher**Day two is about making sure those models can run reliably with new data, identifying issues before they impact customers. It’s also about ensuring that new team members can make changes confidently. For example, a 23-year-old just out of college should be able to tweak a line of code and deploy it, knowing that the system will catch any problems. That requires solid testing, monitoring, and automation frameworks.

## Implementing robust systems for continuous integration and delivery in data science

**30:55  
Alexey**So, thinking in systems means having a platform with integrated components like regression tests, automated deployment, and monitoring. This setup ensures that changes can be made safely and efficiently.

**31:45  
Christopher**Exactly. It’s about finding problems before they reach production. You need robust CI/CD pipelines, test data reflective of real-world scenarios, and infrastructure as code. If you can deploy quickly with low risk and involve new team members in a way that doesn’t jeopardize production, you’ll significantly reduce wasted time and effort.

## Reducing waste and inefficiency through DataOps processes

**34:13  
Alexey**You mentioned that some waste is inevitable. How do DataOps processes help minimize this?

**35:36  
Christopher**DataOps helps by implementing processes and tools that focus on reducing errors and cycle time. Things like version control, automated testing, and observability are crucial. However, adoption is slower than I’d hoped. Even with more companies using tools like DBT, there’s still a lot of heroism and fear-based decision-making.

## Challenges in adoption and the impact of AI tools like ChatGPT

**39:04  
Alexey**Maybe everyone’s just too busy playing with ChatGPT now!

**39:09  
Christopher**That’s a part of it. There’s a lot of focus on generating things—models, dashboards, ETL code—with AI tools. But, focusing on optimizing the creation process only tackles a small part of the problem. The majority of time is spent on rework, fixing issues, and miscommunication. Reducing waste is where the real productivity gains are.

## Automating deployment and using CI/CD to minimize errors

**42:39  
Alexey**How do DataOps processes help in reducing this waste?

**43:02  
Christopher**It’s about automating deployment, using version control, and having tests that run in development before production. Just using Git isn’t enough; you need end-to-end tests and automated checks. Often, data engineers might use these practices, but data scientists or analysts may not, leading to inconsistencies. The whole team needs to be on board with these practices.

**44:30  
Alexey**That makes sense. Still, it's surprising that more teams aren't using CI/CD and Git. To me, it seems like common sense.

**46:27  
Christopher**It is, but there are varying levels of adoption. Some might use Git and basic CI/CD but lack comprehensive testing or integration with all their tools. Others might have pockets of good practice but not across the entire team. What we need is for data and analytics teams to adopt a more critical view of their processes, as software engineers do.

## Shifting focus from development to production in DataOps

**50:29  
Alexey**You’ve shifted your focus from development to production. Why is that?

**50:31  
Christopher**We found that most teams had built things in production without much consideration for development best practices. It was easier to start by observing and monitoring production systems. We also realized that the senior-most leaders, like Chief Data Officers, often don’t last long in their roles. So we shifted our focus to individual contributors—data engineers and scientists who can start implementing these practices.

## Importance and role of Kubernetes and data versioning best practices

**52:42  
Alexey**A question from the audience: How important is learning Kubernetes in the industry? Has it been widely adopted?

**52:42  
Christopher**Kubernetes is important, but it’s complex. Learn Docker first. If you’re managing a smaller team, you might not need Kubernetes. It’s beneficial if you’re running many processes, but there are lighter-weight options that might work better for smaller use cases.

**54:05  
Alexey**There are also tools like Google Cloud Run and other serverless options that might be simpler to use. Another audience question: How is data versioned in the industry these days, and what’s your advice?

**56:17  
Christopher**I’m not a big fan of versioning data itself. I prefer immutability—keeping the raw data unchanged and versioning the code that acts upon it. Focus on having immutable data with functional access methods and version the processing logic instead.

## The role of mindset and culture in reducing turnover and improving DataOps practices

**58:15  
Alexey**That approach aligns with functional programming principles, where immutability simplifies concurrency issues. Final question: Should the solution for high turnover in teams be more about mindset and culture rather than just tooling?

**58:34  
Christopher**Absolutely. Culture and mindset are critical. Tools alone won’t solve the problem. Teams need to advocate for better processes and leadership needs to prioritize building systems that reduce frustration and increase efficiency. It's about making work more enjoyable and sustainable.

**61:20  
Alexey**We could keep discussing this for hours, but we’re out of time. Chris, thanks for joining us so early in the morning and sharing your insights. I really enjoyed our conversation. Thanks, everyone, for tuning in. Looking forward to catching up again in a couple of years.

**64:04  
Christopher**Thanks for the opportunity. I enjoyed it. Take care, everyone!

**64:07  
Alexey**Goodbye!