SDS PODCAST EPISODE 825: DATA CONTRACTS: THE KEY TO DATA QUALITY, WITH CHAD SANDERSON

Jon Krohn:	00:00	This is episode number 825 with Chad Sanderson, CEO of Gable.ai. Today's episode is brought to you by epic LinkedIn Learning instructor Keith McCormick, and by Gurobi, the decision intelligence leader.
	00:20	Welcome to the Super Data Science Podcast, the most listened to podcast in the data science industry. Each week we bring you inspiring people and ideas to help you build a successful career in data science. I'm your host, Jon Krohn. Thanks for joining me today and now let's make the complex simple.
	00:51	Welcome back to the Super Data Science Podcast. Today we've got Chad Sanderson, a brilliant entrepreneur and extremely smooth communicator of technical information as our guest on an episode of the show dedicated to data contracts. Chad is CEO and co-founder of Gable, a platform for data teams that has raised $7 million in seed funding. He's also chief operator of the nonprofit Data Quality Camp. He's author of the forthcoming O'Reilly book Data Contracts and his informative social media posts on data contracts have enabled him to amass over 80,000 followers on LinkedIn alone.
	01:27	Today's episode will appeal most to folks who work with data hands-on or who are involved in management roles that oversee data flows. In today's episode, Chad details what data contracts are, the critical concept of "shifting left" in data quality and governance, how data debt accumulates and leads to spaghetti data architectures, and why data quality is fundamentally a change-management problem. All right, are you ready for this important episode? Let's go.
	02:01	Chad, welcome to the Super Data Science Podcast. I'm so excited to have you here on the show. Welcome, where are you calling in from?

Chad Sanderson: 02:08 Hey, Jon. Thanks for having me. Well, right now I'm calling in from the Cotswolds in the UK, but normally I'm based out of Seattle.

Jon Krohn: 02:17 Nice, it is beautiful in the Cotswolds and I guess you can typically enjoy there your cloudy raininess that you probably love from Seattle.

Chad Sanderson: 02:27 The weather is not terribly different from what I'm already used to, so it can't ruin my day at all.

Jon Krohn: 02:32 Nice, so you were recommended to me as a guest almost a year ago now by Emily Pastewka, who was on the show in episode number 749. So that was in January of this year that it came out, and she highly recommended you as a guest. I've been following you online for some time now and I'm glad that we finally both had an opportunity to get you on. So thanks to Emily there. So you are the CEO of Gable, which is a data contracts platform, and you're writing The Definitive Guide to Data Contracts with O'Reilly, probably the most prestigious technical publisher that you can be writing with for our space. So tell us about data contracts. Your book introduces them as a solution to the persistent data quality and data governance issues that organizations face, but candidly, it's not something that I had heard much about. When I first saw that that's what you were expert in, I was thinking about Web3 or the blockchain. It somehow sounded like that kind of contract to me, but I don't think it has anything to do with that.

Chad Sanderson: 03:39 That's right. So one of the big problems that has manifested itself in the last 10 or 15 years or so, really since the cloud took over as the primary place that companies are storing massive amounts of data is that back in the old days, you used to have a producer of data and a consumer of data that were very tightly connected to each other and more of a centralized team that was

thinking about the data architecture and which data is actually accessible and could be used by a data scientist or a data engineer or an analyst, and they went through a lot of time and effort to construct a highly usable, highly semantically representative data model.

04:21 But now thanks to the internet and thanks to the cloud, you've got so much data flowing in from everywhere, from hundreds, tens of different sources and when things change, it causes lots of problems for anyone who's downstream of that data for models, for reports, for dashboards, and things like that. So the data contract is starting to adopt a lot of the similar terminology and technology as software engineers who use APIs, which is effectively a service contract. It's an engineer saying, "Hey, this is what my application produces. You can expect this not to change. Here are some SLAs around that service." And you can trust that there's always going to be a certain level of latency and uptime, and we're taking that approach and applying it to the data as well.

Jon Krohn: 05:06 So it is similar in software engineering to the idea of, what is the term of software engineering? It's like a service contract?

Chad Sanderson: 05:12 Service contract, yeah.

Jon Krohn: 05:15 And so you're taking those kinds of ideas from software engineering, applying them to data space?

Chad Sanderson: 05:23 Yeah, exactly. Data is obviously very different from applications. You need to think about the number of records that are being emitted at any particular point in time. If a team always expects there to be a thousand events in an hour and in one particular hour, it's one event or two events, that's definitely a big problem. The schema matters a lot. If you suddenly drop a column or add a new column that's an incremental version of a

previous column, it's a really big deal. If you change the semantic meaning of the data, this is obviously another really huge deal. If I've got a column called distance and I as the producer have defined it to mean kilometers, but then I change it to miles, that's going to cause an issue. So the same sort of binding agreements that APIs have, sort of the explicit definitions of expectations coming from a producer, we're starting to apply that to the data producers and not just the software engineers on the application.

Jon Krohn: 06:21 Very cool. Sounds really valuable. In chapter two of your forthcoming book, you discuss how data quality isn't about having pristine data, but rather about understanding the trade-offs in operationalizing data at various levels of correctness. So how can organizations strike a balance between data quality and the speed of data delivery?

Chad Sanderson: 06:44 That's actually a great question. So my definition of data quality is a bit different I think from other peoples. In the software world, folks think about quality as it's very deterministic. So I am writing a feature, I'm building an application, I have a set of requirements for that application, and if the software no longer meets those requirements, that's what we call a bug. It's a quality issue, but in the data space, you might have a producer of data that is emitting data or collecting data in some way that makes a change, which is totally sensible for their use case.

07:20 So as an example, maybe I have a column called timestamp and that's currently being recorded in local time and I as the engineer decide to change that to UTC format. Totally fine, it makes complete sense. It's probably exactly what you should do, but if there's someone downstream of me expecting local time, they're going to experience a data quality issue. So my

perspective is that data quality is actually a result of mismanaged expectations between data producers and data consumers, and that's sort of the function of a data contract is to help these two sides actually collaborate better with each other, to work better with each other and not so much prevent changes from happening.

Jon Krohn: 08:00 So when you talk about data producers and data consumers like you just did there, is that typically referring to internal in an organization or I guess it could equally apply to an external facing API?

Chad Sanderson: 08:12 Exactly, so a producer or is really anyone that is making a unique transformation of the data in some way, which could mean the creation of the data itself. That might be an internal software engineer who is creating an event that's emitted from a front end, like a user clicks on a button in a web app. It could be someone who, a DBA who owns a database. It could be a data engineer who's aggregating all of that data together and creating a silver and bronze and gold data models. It could be a data scientist who aggregates all of this into a training set that ultimately another data scientist in the company ends up using. It could be a tool like Salesforce or a CRM or SAP for an ERP, or it could be someone outside the company altogether, like another company providing an API or an FTP, sort of data dump or something like that. The problems are the same regardless.

Jon Krohn: 09:08 Can you break down for us, as we've now been talking about data contracts, I get the utility, but can you break down for me what they look like? How is it formatted? How do you share it and how does somebody receive it? How do they read it?

Chad Sanderson: 09:27 Yeah, so this is where data contracts are a little bit different from the service contracts where you have something like an open API standard. In the data contract

world, it's more about having a consistent abstraction and then being able to enforce or monitor that abstraction in the different technologies where data is created or moved to. So I prefer using something like YAML or JSON to describe my contracts, and it has various components within it. So you might lay out the schema, the owner of the data, the SLAs, the actual data asset that is being defined or being referenced by the contract, any data quality rules, PII rules, and so on and so forth, and then the goal is to translate all of those constraints into monitors and checks against the data itself as it's flowing between systems or potentially even before that data has been produced or deployed in some way, but I've seen teams that have rolled out data contracts as [inaudible 00:10:30] pages as Excel spreadsheets. Really anything that allows a producer to take ownership of a data asset I think works as a first step towards data contracts.

Jon Krohn: 10:41 Awesome. Yeah, crystal clear. Let's talk about trustworthiness around data. So we've talked now about data correctness, which relates to trustworthiness, and so you've argued that the value of data hinges on its trustworthiness. So how do data contracts help establish trust between data producers and consumers? And what role do data contracts play in rebuilding trust if it's been lost?

Chad Sanderson: 11:09 So I think trust comes down to a couple components. One component of trust is understanding, and the second component of trust is meeting a consistent expectation. And when I say understanding, what I'm referring to there is I am more willing to trust a data source or a data set if I understand what it actually represents. When a table is called customer orders, does that mean customer orders that were placed through our website or through our application or through both or through our customer service line? Does it just refer to a certain type of customer or a certain type of order? So the more

information I have about that data asset, the more that I can actually trust it, and then the second part of trust is the expectation setting. So what is going to happen to that data set over time? Is it going to be changing every month?

12:05 Am I going to know when it changes? Will I know the context of the change so that I can adjust my training data or my query? I think the same is actually true in real life, right? If someone says to you, "Hey, Jon, I'm going to be coming over to your house later, but I might be 30 to 45 minutes late because of traffic." You'll respond very differently than if someone is just 45 minutes late and they don't tell you, they just show up. So I think this is where trust comes from and the data contract is really all about setting the expectation and also helping people understand what the data actually means and how they should use it.

Jon Krohn: 12:39 Nice, that's a great analogy there with the showing up late. It's exactly it that if you can describe that there's going to be some issue coming up, then people can deal with that or they can at least adjust their expectations accordingly. It makes a lot of sense. In your experience, what are companies most common misconceptions about data quality when implementing AI? Because we haven't talked about machine learning or AI systems yet, so what are company's most common misconceptions about data quality when implementing AI and how can they address these gaps to ensure AI systems provide reliable insights?

Chad Sanderson: 13:14 Yeah, so I think there's a few really big problems. In fact, I was talking to a friend of mine about this about a week ago, and their company spent two quarters with every engineer and every data scientist in the business doing nothing but investing into generative AI, and coming out of that, they had all these awesome applications, but their big problem was quality because when the output of the

model was incorrect, they actually weren't able to delineate whether or not the problem was caused by having the correct data and the model hallucinated or if the data itself was incorrect and the model made the right decision. So being able to distinguish between those two things is actually very hard if you don't have some set of expectations for what the data should look like at every step of that transformation journey. So not prioritizing that as a critical element of ensuring model correctness, I think is huge.

14:11 The other thing that I think a lot of people don't put that much attention on when it comes to artificial intelligence is do we actually have this right semantic meaning of the data that is feeding into our training set? It's very easy to say well okay, I can go and pull together a notebook or I can pull data from something like an S3 or whatever, or a spreadsheet, and I can create this awesome training set and I pipe it into my model and now I have all these amazing predictions.

14:38 But did you actually make the right assumption that the data meant semantically what you thought it did? So for example, if I say okay, I've got some vehicle data and I want to use that vehicle data in order to predict what any customers visiting my website might buy, what if that vehicle data only applies to customers over the age of 55? Well, if the audience on my website is the average age is 24 or 32, that data may not be extraordinarily predictive, and that's not something that you're necessarily going to get by examining the data itself. You need to have context about where it's coming from.

Jon Krohn: 15:18 Keith McCormick, data scientist, LinkedIn learning author, and friend of this podcast, has a new LinkedIn Learning course. In the new course, Keith discusses the critical initial stage of Problem Identification and Solution Design. This is a missing element from virtually all data

science training but if you are in a consulting role, either internal or external, you need the skills discussed in this course. You may know Keith from episodes 628 or 655. Be looking for his return on an upcoming Friday episode. You can access the new course by following the hashtag #SDSKeith on LinkedIn. That's #SDSKeith. Keith will share a link today, on this episode's release, to allow you to watch the full new course for free. Keep an eye out for more course links from Keith in the coming weeks.

16:07 Yeah, that makes a lot of sense. It is funny how, this is off on a bit of a tangent, but in the last year or two companies putting so many resources into generative AI systems, believing that there's kind of an infinite amount of value there to users when it's only valuable to your users if the underlying data are providing something valuable to them. So it doesn't matter how great the LLM is if the data that they have access to through your system isn't particularly special or useful.

Chad Sanderson: 00:16:43 Yeah, it has to be valuable. It has to be correct, it has to mean something. Ideally, it should be timely. There should be a use case behind it. It's not really possible to just take as much data as you possibly can, throw it into a model and cross your fingers and hope good things happen.

Jon Krohn: 00:17:03 With your role as the chief operator at the Data Quality Camp, which is, so that's Data Quality Camp capitalized, that's like the name of the organization. It's an online community interested in data quality. You've seen firsthand the growth of interest in data quality through your leadership of that organization. What key insights have you gathered from this community or maybe how have things changed over time?

Chad Sanderson: 00:17:29 So I would say that Data Quality Camp is the largest online data quality community on the internet. It's a

Slack group has over 10,000 members. That really interesting thing I've observed over the years of managing the community now is that people are starting to take data quality very seriously at the enterprise level, and that's because large companies are starting to move beyond the first stages of data management maturity, and I usually break this down into three stages where stage one is, look, we're just getting started with our infrastructure. We need somewhere to put data so that our data science organization can easily access it. We've got to be able to get that data to them on some schedule, has to be timely, has to be fast, that's all kind level one infrastructure level, and then what happens after you get that system into place, things start going wrong, your model starts making the incorrect predictions, you start having change management issues where the upstream systems are evolving in ways that cause problems for the downstream systems.

00:18:34 You have people misinterpreting what the data means. You have people accessing data that they probably shouldn't, but you don't know what the blast radius of those problems are. So you begin to layer in things like observability, that's just checking, well how frequently do we see a problem? How frequently do we see anomalous behavior with our data? And the third step now that's becoming a lot more popular is okay, well once we know how often a problem is happening, what do we do about it? How do we actually prevent it from happening ideally in the future, but at minimum, how do we root cause it much faster? So it's not taking three weeks or four weeks to actually solve these problems once we detect they exist. That's probably the biggest change I've noticed over the past few years.

Jon Krohn: 00:19:16 Nice. That's a nice overview of how things have been changing there in the Data Quality Camp online community, and I guess if any of our listeners are

interested in joining that community, it's super easy to do that, I guess?

Chad Sanderson: 00:19:27 Yeah, just Google Data Quality Camp Slack, it'll pop up.

Jon Krohn: 00:19:30 Nice, we'll be sure to have that in the show notes for people to sign up for. Briefly, because I realize that this isn't your job. We'll be talking about Gable in a second, which is actually your day job, but as chief operator at Data Quality Camp, what does that mean to be chief operator? The term operator to me is kind of the way that investor is kind of distinguished from operator in an entrepreneurial setting. So the operator is somebody who's actually doing something in an entrepreneurial organization with something like the Data Quality Camp. How did you come to that title? Chief operator?

Chad Sanderson: 00:20:11 Well, it's not a business, so I couldn't really put CEO, but I am the main one who created it and the main one who is doing things. So I felt that chief operator was maybe the best descriptor I could come up with.

Jon Krohn: 00:20:23 Nice, I like it. I haven't seen it anywhere else before, but it is descriptive and accurate, so nice.

Chad Sanderson: 00:20:30 Thank you.

Jon Krohn: 00:20:32 In a Substack article that you wrote recently about culture, you mentioned that many traditional data management tools are reactive rather than preventative, and so I'm going to use this Substack article as a jumping off point into Gable, your company. So how does Gable, your company's approach to data contracts shift this paradigm so that we're yeah, no longer being reactive and we are now starting to become preventative with tools like Gable? Yeah, how does that work?

Chad Sanderson: 00:21:05 Yeah, so there's a new paradigm in data management and data quality and data governance that's been kind of bubbling within under the surface for the past couple years or so called "shift left", and this did not start in data actually, it started in the security space, and what security engineers realized maybe five, seven years ago was that the only way to prevent security incidents and hacking and fraud and so on and so forth was to shift all of the security management best practices into the actual code base where the software engineers were doing their daily work. Otherwise, security was always going to be an afterthought to the application teams who are writing code and the security teams could only respond reactively to when they were hacked or when they did detect the data quality issue or a security issue had occurred, and data, I think is following very closely behind security in the same way.

00:22:07 Data quality issues are not something that data producers are thinking about actively, and that's because the data that a producer is creating is normally used for a very different reason than what a data scientist is using it for. So if I'm a software engineer and I own an operational database, I'm using that operational database to run my service, to run the app, but that data, it could be customer data, it could be purchase data or item data, becomes incredibly useful for training machine learning models for doing analytics and so on and so forth, and so the question is how do you get a software engineer who is not thinking about machine learning in their day-to-day job to start taking it seriously? And the only real way to do it is to push the requirements of data quality and data governance into their development workflow.

00:22:58 So like I said, this is what security is doing and this is what Gable is starting to do for the data management and the data quality space. Seamlessly, check for data quality issues, data governance issues when data producers are

writing code, committing code, generating PRs, and helping the teams who use that data understand the changes that are coming before they manifest into a production environment. So to just give you a quick example, I mentioned that you might have a software engineer who decides to change a timestamp column from local time to UTC. What Gable can do is we can check that code before it's actually deployed into production and say, "We know that Jon the data scientist downstream is using that local time data in their machine learning model."

00:23:50 And we can provide that feedback to the software engineer, "Hey, wait a second, there's a machine learning model that is dependent on this data. You shouldn't make this change until you talk to Jon." And at the same time, we can give Jon information that says, Hey, there's a change that's coming that's going to impact you. Now is the opportunity to either get in front of it and have a conversation with the person who's deploying that change or update your training data so you don't get broken. That's the core of what we do.

Jon Krohn: 00:24:15 Nice. This idea of "shifting left" I feel like is something that we need to dig into in more detail. It isn't something that I'd come across before looking into you and Gable. So I mean this is the mission of Gable is this idea of emphasizing shifting data quality, data governance, data management, left. So what is this continuum that goes from left to that we're left on?

Chad Sanderson: 00:24:41 Yeah, so if you think about a normal data pipeline or your average data pipeline, right? Let's just talk about internal to a business for now so it's simple. You've got some application code, some software engineer, maybe a front-end engineer who's writing events. Every time a user logs in, we collect some user data, we update a database, so data flows to a database. Once a user actually interacts

with the website and from the database, we're now pushing that data usually into some storage environment. We might be doing that in batch. We might be doing it through a streaming system like Kafka. So maybe it flows through a Kafka topic into S3, it lands in a file like a CSV or a Parquet file, and then from there you've got your data platform team in most large companies that are picking that data up and they are moving it into the analytical database like a Snowflake or a BigQuery, or they might be using a Databricks or something like that.

00:25:38 And then once it hits the analytical database, then you've got data modeling tools that are used like DBT. You've got orchestration tools like Airflow that are responsible for sort of shuttling the data around, and you might have multiple different data models and multiple different transformations, one, two, three, four, five, all the way up to 10 or maybe even more than that before the data ultimately arrives to a consumer who can actually use it, and then after that, you've got the actual data product. So you've got the data asset that is delivering some value for the business, and that might be the training set for your model. It might be the dashboard, it might be the report.

00:26:17 So you have this really sort of long complex chain of technologies and people that are handing data off all along the way and you can see what sort of problems could be caused just by lack of communication if someone at one point in the chain decides to do something that doesn't jive with someone else at another point in the chain. So when we're talking about shifting left, what we're saying is we're taking the responsibility that usually comes from the data engineers and the data scientists and the data platform teams once the data arrives in an analytical ecosystem and applying the governance there and starting to move it closer and closer to the person who's actually producing the data, and that gets you that full end-to-end coverage for data quality.

Jon Krohn: 00:27:01 Makes perfect sense. I get it and I feel like I maybe should have got it before you even explained it now that you have, but yeah, this idea of information flows when we write it out, we typically have them running from left to, and so the idea here is that you're shifting data quality data governance, data management left towards the data producers and further from the data consumers. Nice. Yeah, it seems crystal clear when you do that, how that can foster visibility, accountability, ownership, and all those key things, trust.

Chad Sanderson: 00:27:32 Exactly, one of the interesting things is that in the data world, we're actually pretty late to this concept. It's been around for a really long time in other areas, like QA for example. So I don't know what the average age of your listeners are, but if you've been in tech for a very long time, you probably have worked for a company that had a dedicated QA team, and that's really all they did is they checked the code that some software engineer has written to make sure it doesn't have any bugs, but if you're working at a more modern startup or just a more modern company, you probably don't see that as much. The teams responsible for doing the QA are the software engineers who are actually building the application. So QA has effectively shifted left, same for version control and infrastructure monitoring and so on and so forth.

00:28:18 And the reason why that happened I think is pretty obvious, but if you have a centralized team and all that centralized team is responsible for is quality, you're effectively almost putting them in competition with the teams who are responsible for shipping code and delivering value for the business, and who's going to win? If the quality team who says, "Hey, you've got a lot of problems, you've got issues with your code, you need to fix them." Or the team that's shipping that code and saying, "Hey, if I ship this, we're going to make a whole lot of money." Well, it's obviously the team who makes the

money is almost always going to take priority and what that's resulted in overtime is just a bunch of low quality code that creates a ton of tech debt, a ton of data debt and so on and so forth, and so these folks have sort of realized, well, if we want to have high quality code, if we want to have sustainable infrastructure, then the teams who are making the money also need to be the teams that are thinking about quality.

Jon Krohn: 00:29:12 In a recent episode of this podcast, the mathematical optimization guru Jerry Yurchisin joined us to detail how you can leverage mathematical optimization to drive commercial decision-making, giving you the confidence to deliver provably optimal decisions. This is where Gurobi Optimization comes into play. Trusted by most of the world's leading enterprises, Gurobi's cutting-edge optimization solver, lightweight APIs, and flexible deployment simplify the data-to-decision journey. And, thankfully, if you're new to mathematical optimization approaches, Gurobi offers a wealth of resources for data scientists, including hands-on training, comprehensive Jupyter-notebook examples, and extensive, free online courses. Check out Episode #813 of this podcast to learn more about mathematical optimization and all of these great resources from Gurobi. That's Episode #813.

00:30:00 Yeah, this sounds similar to something that you talk about in chapter one of your book where you talk about how the classical garbage in garbage out cycle that we have on data teams, this expectation that somehow a model, whether it's a machine learning model or statistical model, whatever, some analytics process that's somehow it's going to have magical capabilities. It's going to do exactly what you want even if the data that go in are crap, are garbage. Yeah, so it seems like it's related to that. Why do you think that data professionals today still struggle so much with this problem of garbage data quality given the availability of so many modern data

management tools out there? I guess another way of me asking this question is if we have all of these kind of data management tools out there, why don't they fit the bill? Why can't they fix this data quality problem?

Chad Sanderson: 00:31:00 Well, the short answer is because it's not actually a technology problem, it's a culture problem, and once you've accepted that it's a culture problem, then you can use the most ideal and optimal technology to help you solve it faster and easier. But just to go back, I sort of mentioned this during our intro, but the data world has been changing very quickly over a relatively short period of time, and when technology evolves to solve some problem, it usually ends up creating a new problem as well. And then people try to solve that new problem and creating it then solves another problem and so on and so forth. And in the 1970s and 1980s, I mean data was everywhere. Everybody was using data. People were using data before they were even using software by many years actually, but it was extraordinarily expensive. You had on-prem databases like Vertica for example, and because compute and storage were not decoupled, it cost an enormous amount to store any amount of data to transform any amount of data and to iterate on any amount of data.

00:32:12 So because the amount of data that was accessible to teams was so small, you manage data in the same way that a librarian might manage books in a library, you can be very thoughtful, you can be a steward, you can be very deliberate about what data you bring in, about what data goes out and how you organize that data, how you structure it, how your entire data model sort of works together, and so it was people and centralized teams sort of sitting at the middle of this and making sure that quality was high, that governance was in place, that accessibility was high, and so on and so forth, but then

we solved that problem and we solved the problem of cost through cloud storage, right?

00:32:53 So cloud storage, effectively decoupled compute and storage and storing any amount of data effectively became free, and businesses thought, well okay, well now that it's basically free to store all this stuff, we should be storing as much as we possibly can because we don't know what's going to be useful or not, and we want to give our data teams the opportunity to find what's useful from this large sort of mass of data that we're collecting from all of our systems.

00:33:22 So you started to see this sort of exponential explosion of the amount of data that was available, and when you're talking about almost an infinitely increasing data in some company, that can't be managed by the same small number of data stewards anymore, the same way is that if you try to have a group of 20 or 30 librarians manage the amount of books that are in Amazon, it would be impossible. There's just too many of them, and so what did companies move to? Well, they move to automated systems in the same way that Amazon sort of is moving to an automated systems and that's more of a federated model where you've got a marketplace where the people who are creating the books are working, the sellers are working directly with the people who are buying the books, and in the same way you've got the people who are producing the data that are working directly with the teams who are consuming the data.

00:34:15 The problem is that the industry hasn't fully accepted the federated model. You still actually have that centralized governance or data engineering team or data team that's essentially in the middle trying to get a handle on all the data that's flowing through to that analytical ecosystem and it's just not working, and all the tools that exist today are more about making those centralized teams more

efficient and more effective than at doing their jobs, than embracing the reality, which is you can't have a centralized team with the massive volumes of data that exists. You actually need to allow the producers and consumers to operate in a federated way, and that just requires better communication and change management.

Jon Krohn: 00:34:57 Nice, that was a great explanation. I love particularly the e-librarian analogy, which made so much sense to me. In your book, Chad, you describe data debt, something that we haven't talked about yet. Data debt as a primary villain in data management. So maybe if you are a book or a novel, this would be the data contract is the superhero, I guess, out to get the data debt villain, and so yeah, data debt can be a huge problem in data management impacting data quality and data trust. So again, this seems like a term that could be borrowed from software engineering where we talk about technical debt in the code that we develop, and you kind of have these lurking surprises that could get us in our software development, and so it sounds like data debt might be similar. How do you define data debt?

Chad Sanderson: 00:35:54 I think it is similar. So the interesting thing is that a lot of the patterns we're talking about, whether it's data debt and tech debt or data contracts and service contracts or even data quality and code quality, all have a relationship to each other, but not because I think that data is inherently very similar to software engineering. It certainly does have a lot of similarities, but there's actually a larger category of change-management regardless of what the system is. We could be talking about applications, we could be talking about data, we could be talking about real world systems, supply chains for example where you've got a farmer that's harvesting wheat, they are handing that wheat off to a manufacturer that is turning it into something interesting and useful

like cereal, and then that itself is being handed off to a retailer to sell to customers.

00:36:58 The supply chains actually have the same exact problems as data pipelines do, which is why I've been calling them data supply chains instead of pipelines for the last couple of years or so, there's still many people who may not potentially speak to each other. There's quality issues that could be introduced, which could be really bad, and there's also contracts. So this is very common in the manufacturing space where a farmer needs to have a contract with the distributor because the distributor needs to specify, "Hey, if you give me a certain quality of grain, I am not going to be able to sell that to my customers who can turn it into something consumable for a human being." So there's a certain quality threshold that has to be set. So all of this around change, around handing off materials and data and code between people I think it's part of this larger category.

00:37:51 That's just what I wanted to preface that answer with, but I do think software is more sophisticated than data when it comes to solving or they have better language when it comes to solving certain types of problems, and tech debt is definitely one of them. So in software land, tech debt is all about we have made certain software choices at particular points in time. We knew when we made those choices that they weren't going to account for the long-term scale of the company or the long-term needs of the company, but we just needed to go fast, and at some point in the future that debt has to be made. You do reach that scale, you do reach those long-term needs, and now you have to address it. So the real potential problem with tech debt is that it slows your business down to a crawl as you need to address all those early decisions that you made before you can move forward.

00:38:45 Data debt I think is a little bit different than that. Data debt is when data changes in some way and the team's responsible for consuming that data, don't correct the underlying systems, but add their existing expectations on top. So we say okay, I've got a user table and I've got a first name field and I've got a last name field and the producer decides I'm going to join that together into a single name column. Well now downstream, I as the data consumer, suddenly all my data is wrong, and instead of going through my entire pipeline and fixing everything to reflect the new world that the data producer has created, I'm just going to add a filter on top of my query that says, "Hey, look, I want you to just slap these two things together. Just do a merge and sort of stick first name and last name together and cram it into one shape."

00:39:45 And what happens over time is you get these really long complicated SQL statements that no one who wasn't there when they were written can possibly understand, or you get massively long databases or massively long tables where you have column after column, and if you were to ask the original creator of that table, what it means, they probably wouldn't even be able to tell you and answer anymore. This is sort of what I call a data debt. When the initial meaning of the data incrementally becomes lost over time, and personally, I think this is worse tech debt because tech debt slows you down, but data debt causes you to lose sort of your grasp on the initial intent of the information being conveyed through the data, and that intent of conveying that information is the entire purpose of using the data in the first place.

Jon Krohn: 00:40:43 Well said. Is something that makes this data debt maybe accrue more quickly or more problematic, this spaghetti data architecture that you've talked about on previous podcast appearances or talks that you've given. Yeah, tell us about the spaghetti data architecture and what the root causes are.

Chad Sanderson: 00:41:03 Well, the spaghetti architecture is really a result of data debt forming over time. So when you have the spaghetti SQL, that's sort of the end state where all the code that you've written doesn't really make sense anymore, it's all over the place. You've got copies of data in multiple places. You've got the same concept that people think looks the same, but it actually means something different and you've got different definitions of things, and that is really a result of unmanaged change. So something I've said on a few on conferences and talks that I give is that if nothing ever changes, nothing ever breaks. And that means that quality is actually a result of unmanaged change and that means the data quality problem is actually a change management problem, and so when you don't have good change management, then changes occur. People react to that change, they react to it in a suboptimal way. You start to build the spaghetti SQL over time and at a certain point, it's so complex, it's so difficult to sort of pierce, it's so opaque that it's very hard to move forward.

Jon Krohn: 00:42:07 Are data contracts a solution that completely in and of themselves solve this spaghetti data architecture problem and the data debt problem?

Chad Sanderson: 00:42:16 I don't think so. I don't think a data contract by itself can solve that problem. I think what the data contract does is that it opens the line of communication between a producer and a consumer in a way that didn't really exist before. In my last role at Convoy, I actually went and talked to about 30 or 40 software engineers and I asked them, "Hey, how can I get you to care about the data that you produce from the perspective of analytics and machine learning?" And that's not to say they were doing anything wrong by not caring about it, but I needed to know and the answer they told me is, "Well, I need to understand where my data is being used. I need to know who's using it. I need to know what they're actually using

it for, and once I have that information, then I can start making better decisions."

00:43:03 And because the data producer, to them, the data is effectively going into a black box. That's your analytical database or your Databricks or your machine learning environment. It's a black box to them and they can't see into it. They don't know what's happening. If they start to see, "Oh, my data is being used in this place and this place over here by these teams, when I make changes, now I know who to talk to." And when you start to open that up, you start to actually have conversations. This is when you start falling into Conway's law. So I don't know if you're familiar with Conway's law, but it basically says that the communication structure of an organization is it ultimately becomes a reflection of its architecture. So your actual applications and your services and your systems and the way people build products and write code is based on the communication patterns that exist in your company, and right now, the communication pattern between the teams that create the data and use the data is non-existent, and that's what the contract opens up.

Jon Krohn: 00:44:03 Ready to take your knowledge in machine learning and AI to the next level? Join SuperDataScience and access an ever-growing library of over 40 courses and 200 hours of content. From beginners to advanced professionals, SuperDataScience has tailored programs just for you, including content on large language models, gradient boosting and AI. With 17 unique career paths to help you navigate the courses, you will stay focused on your goal. Whether you aim to become a machine learning engineer, a generative AI expert, or simply add data skills to your career, SuperDataScience has you covered. Start your 14-day free trial today at superdatascience.com.

00:44:42 Nice, I hadn't heard of Conway's law, but I will be reading about that more and including a link to it in the show

notes because that makes a lot of sense to me that if I got that right or if I get it right, the idea of Conway's Law to repeat it back to you is that the ways that we set up communication in our organization end up determining the quality of the communication in the organization.

Chad Sanderson: 00:45:05 That's exactly right. There's sort of a famous infographic where they basically lay out the communication structures at various companies and then show how those communication structures directly map to the way those companies actually built their software. Like Facebook has more of a mesh style communication structure, and their products with the way that the microservices communicate to each other is more of a mesh. So it's a really fascinating idea, and I think that it affects the data world incredibly, because data, it can't exist in isolation. It's not like software, right? Like a software engineer can go out, they can build an app, they can write requirements, they can deploy their code, and as long as it's not incredibly broken, it will more or less work just fine, and the software engineering industry has been pushing engineers to become more isolated over time, right?

00:46:05 With microservices and sort of smaller and smaller theme sizes and sort of working on these more individual product components and so on and so forth, but data doesn't work like that. Data can't exist in isolation. Data has to come from somewhere and then it has to be transformed by someone, and then it has to be used by someone, and usually it's not the teams that are producing the data that are also the ones creating the models for it. So if that is the case, then it's so critical to create that information bridge and allow these teams to work together better.

Jon Krohn: 00:46:38 So we've talked a lot today so far about the present of data quality and solutions that exist today. As AI systems

become more autonomous going forward, and that's happening really rapidly, and so we will have more and more AI systems in our data pipelines the whole way from left to right, generating data, assessing the quality of data, work with data, building software systems, training machine learning models. It's going to be more and more automated and fast. So as AI systems become more autonomous in data processing and data decision making, how do you think the role of human oversight in ensuring data quality will evolve?

Chad Sanderson: 00:47:22 Yeah, that's a great question. So I think that as long as we're talking about AI in its current form and not AI as some master overlord of the human race of everything.

Jon Krohn: 00:47:37 Yeah.

Chad Sanderson: 00:47:38 There are a few things that humans will always have a bit of an advantage at and the main one is understanding context and understanding the meaning of data in the real world. At least the data that we work with in sort of bites and bits is just an abstraction of reality, right? My example I've been using a lot as someone goes onto a website and fills in a form, the data we collect on that event is just a representation of that real world event actually happening, and so someone needs to understand what that real world event is. It can't solely be represented by the data itself because the data only contains a fraction of the information that's required to communicate that sort of instance of reality effectively. So when you're pushing data into a model, a human is making a lot of assumptions that we are assuming that, like I said before, a customer refers to something specific in the real world, a certain type of person, and I think the role of human beings is going to be focusing more on that, understanding how does the actual business work?

00:48:42 How are we representing the business through data? How do we make sure that these definitions and the semantics we use are actually consistent so that we can give that data to a machine and let it process more efficiently? The second thing I think that people are going to need to be able to do is set up the constraints and the rules and the policies for these artificial intelligence systems to actually function. Today, most data producers, irrespective of AI or anything else don't know where their data is actually going, and like I said, that means they can't make optimal decisions regarding their data. So if you have an AI that is trying to optimize a database or optimizing code that produces data without understanding what the downstream dependencies are, you might actually get a much worse situation for the data scientists or the consumers on the other end because changes are going to be happening much faster and those changes are happening potentially in a vacuum.

00:49:41 So there has to be some way to create that bridge to allow the AIs to understand, here are the ramifications of my choices. If I do this, yes, it might be a good decision for this local system, but what is it going to do to the rest of the business? And is that going to be a good thing? And if I do make that change, how do I communicate that to potentially the other sort of AIs and the agents that are roaming about? So I think there's going to be a few set of really interesting challenges there. I don't think that there is a clear solution that exists today on how you even manage data quality or manage data at all when it comes to unstructured data, really everything that I've been talking about is primarily for structured data. I think unstructured data is probably five to 10 years behind, and we're not even doing a good job on the relational data yet. So there's still a lot of work to be done in this space.

Jon Krohn: 00:50:33 Nice point there at the end. That actually clarifies a lot of everything we've been discussing so far in this podcast

episode that you could think about everything we've been talking about in the context of a relational SQL database or a spreadsheet, a flat two-dimensional piece of set of data where columns represent specific things. If it could be typed, it could be relatively well understood, but yeah, especially the generative AI world that we're moving more and more into that allows natural language interfaces with human users or natural language outputs to human users. This generates huge amounts of unstructured data and potentially really vast amounts washing out any of human-generated stuff. So yeah, all kinds of questions there around data quality.

00:51:24 Let's jump now to your background, which is really interesting, Chad. So you have a background in journalism, you study journalism as your degree, and this shaped your approach to data-related leadership emphasizing the importance of clear communication and storytelling, and I've got to say, so you and I were talking about this during a break in recording, but now I'm going to say it on air that your communication is outstanding. It is really exceptional. I mean I do 104 of these episodes a year, and you're absolutely top tier in your ability to convey complex concepts clearly, succinctly, and where possible, you also have some data storytelling in there, some good analogies that make it easier for us to understand like the library example.

00:52:14 Well, it seems clear to me that your journalism background has probably impacted your ability to lead well, particularly your data leadership, and so you've often spoken about the muddy relationship between data producers and data consumers. How can data teams leverage say storytelling techniques to bridge the communication gap and create more effective collaboration between data producers and consumers?

Chad Sanderson: 00:52:43 Yeah, interesting question. So I think that storytelling is really at the heart of all forms of collaboration and the data producer, data consumer relationship like you mentioned is no different than that. The most important thing I learned from my time in journalism is the importance of asking a good question, and you've obviously learned that as well with the podcast, but if you're good at asking questions, you're actually sort of hearing open the abstraction of a person's soul, right? You're starting to understand them more as a human being instead of making assumptions and sort of inferring what their intent was when they communicated some idea to you, and that's why most of the problems that I've solved in my life has come as a result of me asking a lot of questions and trying to understand the day-to-day experiences of a group of people I may not have myself.

00:53:44 So I mentioned doing this for the software engineering team where I said, "Hey, how can I get you to care about data?" But I've done the same thing, not only did I do this within the company that I worked at, but I did it across maybe a thousand other companies from huge tech businesses like Twitter or X now I guess, and Google and Meta to very early stage startups to more of your growth stage companies, and each one of those people had a slightly different opinion and those opinions all helped me frame the problem. It's sort of like there's this big unclear nebulous issue that no one had quite defined particularly well, but everybody understood their own little piece of the puzzle, and the more of those questions got filled in, the more it became clear what the actual problem was and once I understood that problem, that's when I started writing about it online.

00:54:35 But the same strategy, you can imply internally as well, right? What are the things that your data producers need to understand in order for them to take ownership, take accountability of their data, and that requires talking to

them, right? What are the things that they're doing every day? How do they think about data today? How do they think about making a change? How do they think about the data science team and the data engineering team? What do they believe their role in that relationship actually is? And you've got your IC engineers who obviously will have an answer for you, but then you've also got the next layer up at the managers and then the directors, and then the executives of the company, and once you start to get the full profile of how your upstream organization thinks, then the right information to communicate to them to get what you want, which is quality and great data management and great data governance, and you can use this same technique to do a lot of things.

00:55:30 You can do it to sell data science projects for example. I say this all the time, but the metric that I am most proud of in my entire career when I wasn't an entrepreneur is that I had, at least in my last two jobs, I had a 100% success ratio when presenting new projects to my leadership team. I never got turned down once, and the reason that happened was because I was exceptionally good at selling those products to an executive, and the reason I was very good at selling is that I was very good at asking questions and understanding what their needs were and what their pains were, and then being able to tie my initiatives to that, to solving their problems.

Jon Krohn: 00:56:13 Very nice. I love that. That is tremendous, very cool. It's been awesome to hear now your journey kind of toward where you are today and it's been great to hear over the course of this episode the great things that you're doing at Gable that you're doing in the data quality community, in the Data Quality Camp online community, and yeah, as I already said, I've been blown away by how well you answer questions, so it's been a treat to have you on.

Thank you so much, Chad. Before I let you go, I ask all of my guests for a book recommendation.

Chad Sanderson: 00:56:51 So I would recommend Nassim Taleb's book, Fooled by Randomness. It was a life-changing book for me when I was making the decision to switch from journalism into data science, and I was really first starting to learn statistics. His ability to take statistical concepts and map them onto the real-world decisions we make every day was incredible, specifically when it comes to entrepreneurship. So definitely give it a read if you have the chance.

Jon Krohn: 00:57:22 Very nice, thanks for that and then how should people be following you after the show to get more of your crisp, clear insights? I know you have a huge following on LinkedIn for example, over 80,000 people following you on LinkedIn. Is that the place to go, or are there other places to follow you as well, I guess also and your online communities?

Chad Sanderson: 00:57:41 Yeah, so you could definitely follow me on LinkedIn, which is just linkedin.com/chadsanderson. I also have a Substack that I maintain a little less regularly these days because I'm writing a book and also running a company, but it's dataproducts.substack.com

Jon Krohn: 00:57:59 Nice. Well, thank you for taking time out of all of those things that you do, including time away from your partner. I think we owe her some thanks as well for allowing you to record this episode while you're traveling with her in the Cotswolds. I hope you have a wonderful time there and I look forward to catching up with you again in the future.

Chad Sanderson: 00:58:19 Awesome, Jon. Thanks for having me.

Jon Krohn: 00:58:26 In today's episode, Chad filled us in on how data contracts are formal agreements between data producers and data consumers defining expectations for data quality, schema, and semantics. He talked about how "shifting left" in data management means moving quality checks and governance closer to data production, improving overall data reliability. He talked about how data debt accumulates when changes to data aren't properly communicated or managed, leading to complex hard-to-maintain spaghetti data architectures, and he talked about how data quality is fundamentally a change-management problem requiring better communication between data producers and consumers.

00:59:04 As always, you can get all those show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Chad's social media profiles, as well as my own at superdatascience.com/825, and if you'd like to connect in real life as opposed to online next month, I'll be giving a keynote and hosting a half day of talks at Web Summit. It's coming up on November 11th to 14th in Lisbon, Portugal. With over 70,000 people there, I'm pretty sure it's the biggest tech conference in the world. It'd be cool to see you there amongst all those people and meet up.

00:59:41 Thanks of course to everyone on the Super Data Science podcast team, our podcast manager, Ivana Zibert, media editor Mario Pombo, operations manager Natalie Ziajski, researcher Serg Masis, writers Dr. Zara Karschay and Silvia Ogweng, and founder Kirill Eremenko. Thanks to all those great folks for producing another important episode for all of us today.

01:00:01 For enabling that super team to create this free podcast for you, we are deeply grateful to our sponsors. You can support this show by checking out our sponsors links, which are in the show notes, and if you yourself are

interested in sponsoring an episode, you can get the details on how by heading to jonkrohn.com/podcast. Otherwise, share, review, subscribe. Any of those things are awesome, but most importantly, I just hope you'll keep on listening. I'm so grateful to have you listening and hope I can continue to make episodes you love for years and years to come. Until next time, keep on rocking out there and I'm looking forward to enjoying another round of the Super Data Science podcast with you very soon.