

**EPISODE 1178**

[INTRODUCTION]

**[00:00:00] JM:** Data science is a collaborative field. Collaboration requires sharing the artifacts that data scientists are working on such as Jupyter Notebooks and SQL tables. Hex is a platform for improving sharing across data science workflows. Caitlin Colgrove and Barry McCardel are founders of Hex and they join the show to discuss what they've built.

[INTERVIEW]

**[00:00:26] JM:** Barry and Caitlin, welcome to the show.

**[00:00:28] BM:** Thanks for having us.

**[00:00:30] CC:** Yeah, great to be here.

**[00:00:32] JM:** You both worked at Palantir, and Palantir was started sort of before tooling for data tools really got good. What did you see at Palantir in terms of the evolution of data tools?

**[00:00:47] CC:** Yeah. So that's actually a really interesting question because I think there's been a lot of movement here just during my career. I've been working in data and data integration for many years now and I think when I started – And what we saw at Palantir was that most companies had actually surprisingly limited capabilities here. You would go into big companies and their data would be all over the place. Smaller companies were trying to do the right thing but didn't have the bandwidth to really build out a reliable set of data infrastructure. And then over the last five years you've really seen I think a significant shift in where the industry is. So you have things like Segment and Fivetran to help manage your data flow. You have Airflow and Transform and DBT to do a bunch of your pipelining, and then all of this is landing in Snowflake, which is a really easy and scalable data warehousing solution.

And so now actually what you're finding is Palantir would really excel that was helping people get all their data in order so that they could even begin to do something useful with it. And now we're really coming into companies and finding that they have a lot of this stuff already in place, but they're still kind of stuck on that last mile question of, "Okay, well we have all this data and we really want to be a data-driven organization. How do we actually take that data and turn it into something that's useful and accessible in particular to the rest of the people at the company?" And I think that's honestly where Hex can have some of the biggest impact.

**[00:02:17] JM:** How did those observations lead to the creation of a product?

**[00:02:23] BM:** Yeah. I think as Caitlin mentioned, sort of we've seen the landscape and the data world evolve a lot over the last few years. And one of the big observations we've made, Caitlin talked about this this world where organizations are more frequently now having their data in sort of a usable format. One of the big observations and the thing that kind of led to Hex was you've seen this growth in the data space. You've seen a growing number of data teams and these organizations that want to be really data-driven. Lots of new data analysts and data scientists coming in. These are some of the most in demand roles. The data is now there for them to do interesting things and they're really interested in how they can take that data and create impact and ROI within the business and not just be like an R&D function where they're just doing interesting sort of research.

And after spending a lot of time in the space and seeing a lot of different data teams and how they operate, one of the biggest gaps that we identified is sort of around this sharing and collaboration, but you see a lot of data workers that have a hard time both working with each other. So like creator to creator collaboration. But I think more importantly how they're able to connect that work to the rest of the organization and connecting what they're doing to the questions that they're trying to help answer for the business they're in or presenting it to customers. And so I think that's really where we're focused and what we are most interested in is we're building Hex is sort of how to make those sharing and collaboration workflows in the data space really awesome and something that is supported really well in a first-class way.

**[00:03:56] JM:** What are the frictions around data sharing that aren't handled simply by Jupyter Notebooks?

**[00:04:03] BM:** Yeah. So Jupyter Notebooks are awesome. I've been using them for a long time. I think the Notebook format is great for a lot of things. And we see a lot of data teams that have sort of adopted these as their way of iterating and working on the logic and the analyses they're doing. I think the classic thing we see though is the last file of that winds up getting really painful. So the story is like – The sort of classic workflow we see is I'll pull my data, usually pulling it locally like I'm working on my notebook, running it locally. I'll iterate on my logic. I'll get it into a good place, whether it's an analysis a building or a model or sort of a deep dive on a question I've been asked. And then if I want to actually take that and share it with others, I wind up sort of falling back to these very like legacy ways of doing that. So I might be screenshotting charts. I might be downloading sort of CSVs of the data and sharing them that way. Taking these things, pasting them into Google Docs or slides, or very often we see people sort of creating these Rube Goldberg machines where they're taking their notebook and trying to get run on a cron job so it can dump results into a database so we can look at it in our BI tool. And it's like these are very common sort of friction points around taking the work people are doing and actually making it accessible and useful to the rest of the organization. And we just think there's a lot of overhead there and there's a lot of interesting room to explore I think in how you can help data teams better connect their work to the goals of the organization and make them feel more impactful.

**[00:05:38] JM:** Did the problems that you experienced at Palantir, are those reflective of problems that are affected by the broader industry?

**[00:05:48] BM:** I think so. I think the problems – As Caitlin mentioned a little earlier, I think a lot of the things we saw at Palantir were around data integration and how you can take data from a lot of different sources. And this is what Palantir always excelled at whether it was in the government space or the commercial work we did. And I think that is sort of a very evergreen problem. And it's been very cool to see a sort of ecosystem of new tools popping up to help solve these things. And I think data integration is a surprisingly broad and deep problem. And I

don't think it'll ever truly be solved, but it's cool to see a lot of new tools popping up that I think make it much more realistic for teams to get their data in great usable shape.

And I think it's very interesting to see for early stage companies, ourselves included. Even just five years ago if we wanted to set up a reasonable data stack for us to be able to integrate and utilize a lot of the data we're collecting ourselves, be it data from our marketing site or product analytics or whatever, that could have been like a multi-week journey of trying to get all of that set up at the least. And today you have a lot of tools that just make it much more seamless and frictionless to get your data in a reasonable shape. In something like a Snowflake or a Redshift, that makes it very easy to query and access. And now I think the questions are as more and more companies have that sort of data layer built out and they're layering on things like data quality tools or data catalogs to sort of ensure that the data is discoverable and in good shape, what are you doing with it? How are you making that useful and leveraging it in the business? And I think those are some of the questions that we certainly saw at Palantir and I think a lot of companies are now getting to.

**[00:07:31] JM:** So talking more intimately about Hex, can you describe the workflow for using the Hex product?

**[00:07:39] CC:** Yeah. So our goal here is to have a workflow that's the fastest way to go from your analysis to something that's interactive and shareable. And so really you start it the way you would start any analysis. You start with your data. So we have a bunch of really nice features around how to connect to your data, whether it's built-in support for a lot of popular databases, secure ways to store your credentials. This is one big pain point that I think Hex can provide a lot of added value to. We have a first class SQL editor. So all of these things really to just help you get your data into your analysis as fast as possible.

And then from there we have a Python-based notebook UI, and this is something that a lot of people use. It's really popular. It was made popular by things like Mathematica and Jupyter. We've already talked a little bit about Jupyter. But for those not familiar who are listening to this, basically it breaks down your code into cells that you can execute independently and

iterate on and the outputs are just right there in line. Super popular because it's a really fast, really nice way for people to iterate on their data analyses.

Though there is some debate here about whether or not notebooks are necessarily the right tool, but we personally love them. We've had great experiences with them. And more importantly we really want to meet people where they're at and really give them the tools that they're already comfortable with. And this is one reason why we're 100% compatible with a lot of these open source tools right out of the box so that people can just bring in their existing work and get started right away.

From there, Barry talked a lot about the collaboration side of this. So some of that is your typical kind of code-based collaboration, whether it's people working on the notebook at the same time, leaving reviews, publishing, things like that. And some of it is more around kind of the sharing and permissioning side. So who can have access to this it? And really it's going to look pretty similar. If you're familiar with something like Google Docs or Figma, that's kind of the style of collaboration that we have in these notebooks.

From there it's super easy to go and create your data application or other sort of interactive artifact that you want to create from your data. Basically what we do is we allow you to take your existing notebook so you don't have to really rewrite anything and use the cells and the outputs and the interactive widgets and things that are already in your notebook and compose those into UIs. And from there you can control who sees it by these sort of granular role-based permissions. We also keep a history of all the runs of the notebooks. So if you ever want to see what a particular analysis looked like back in time, you can really easily go back and see that.

**[00:10:15] JM:** Tell me more about what happens? Like what the difference is the delta between a team that will be using Hex and a team that is not using Hex?

**[00:10:27] BM:** Yeah. So what we see a lot of teams are doing today is they're doing their work in things like notebooks. They're typically working locally. And they're doing really interesting work typically and they're either answering questions for the business or they're developing

models or they're working in an analysis trying to understand a key question that's been posed to them. And again the typical workflow today that we see a lot of people doing is they're taking these. They're trying to collaborate with them through GitHub, which has not traditionally been a great way to manage and collaborate on sort of these notebook analysis-based workflows. And so there's sort of a core pain point there just in terms of making that creator to create or collaboration more seamless.

And so Hex, we have you know real-time collaboration. So you can have multiple people in there looking at the same thing at the same time. We have cell locking. So it's also kind of nice that you're not stepping on each other's work, and then comments so I can leave a comment and you can sort of have these sort of review and feedback workflows directly in the artifact that you're working on. And then I think the biggest delta though and the biggest difference between sort of the traditional workflows we see teams doing and what you're able to accomplish in Hex is taking that work and turning it into something that's useful and usable for the rest of the organization. And so the big difference here would be instead of the output of our work being a bunch of screenshots that we're putting into a deck or extracting things into a spreadsheet that we're sending around or you see a lot of like sort of statically rendered PDFs that wind up getting pasted into Slack channels. I can take my logic and instead turn it into a live interactive web app. And that could take the form of something like a report. So a combination of visualizations and data and commentary around it could take the form of a dashboard that is live and updatable and non-technical people within my organization could come in and add filters or parameters and sort of cut analysis in a different way, or something that's much more sophisticated. And I think there's a lot of power that we expose in terms of being able to take something and even build like a data transformation tool or something that's a little more workflow oriented where I can come in and even be pushing data to other places.

And so it's really taking – And what I think that the thing we're trying to do is help data teams take their work and just turn it into something that's much more usable and much more accessible and impactful for the rest of the organization. And I'd say that's the biggest difference between that we see and the biggest impact we're trying to create for the teams we work with.

**[00:13:01] JM:** So this ecosystem has a lot of tools that are fairly disconnected. We've already talked about some of them, Nnowflake, DBT, Fivetran, Jupyter Notebooks, all these different things. Is it hard to build a product over this shifting landscape?

**[00:13:20] BM:** Yes and no. I think you want to be mindful of how workflows are evolving and the new things happening. And there are certainly some places where I think as things change we may want to make sure we're keeping those things in mind. But in general, I think especially where we are in the stack, you know we're focused on really awesome integrations with the data layer. And so for us that looked like taking the time to develop a really robust data connections backend and a series of plugins that allow us to connect to these common data sources. And so we're a little more agnostic to whether you're using DBT or Dagster or whatever you're using for transformation or whether you're using Fivetran or Stitch for extraction. I think for us, as long as the data is making its way into a common data source like Postgres or Snowflake or Redshift or any of these that we support, you can pick that data up and immediately be productive in Hex.

I think there's a lot of opportunity as we look down the road for how we want to start exposing more functionality through our API, through connecting to other tools that are exposing really robust APIs and enable workflows that transcend just pulling data out of a data warehouse and doing analysis on top of it. I think there's a lot of interesting opportunities down that route, and that's one of those areas where I think we're trying to spend a lot of time listening to our customers and what they would want to do with that.

**[00:14:46] JM:** How does a focus on collaboration create difficulties in developing the product?

**[00:14:51] CC:** Yeah. So this is actually something that we have some experience with even before Hex. So at Palantir we actually worked on some real-time collaboration infrastructure. And so coming in, we knew that this is something that we wanted to incorporate in the product. I think one trend that we've seen largely in the industry with a lot of these new SaaS productivity tools is that the assumption is that they're collaborative by default. If you have two

people coming in and editing the same artifact, it should just work and people are surprised when it doesn't. So this is something that we're thinking about from the beginning, but really real-time collaboration and a lot of the tooling there is still kind of in its infancy in a lot of ways. And so it's a fairly challenging engineering problem.

We came in sort of with an open mind even though we've done it before and did a survey of where the industry is at in terms of standard algorithms for collaborative editing, and there're really two big ones. So the older, more established one is called operational transforms. And if you're familiar at all with Google Docs, this is what powers a lot of that. But there's a new sort of up and coming algorithm in this space called conflict-free replicated data types, CRDTs. And newer products like Figma are based around some of these.

So we kind of did an examination of the two of these and what are the tradeoffs. OT is very well established and it handles complexity fairly well, but the downside is there's actually a lot of boilerplate in implementing it. And it does rely on a centralized server in order to work. Whereas you look at something like CRDT – And there are a lot of advantages. You don't need a centralized server. You can decompose everything into very intelligible single operations on data structures, but a lot of the semantics can get kind of complicated around that. And a lot of the implementations out in the wild today are still somewhat new and untested and often lack a lot of the features that you can build with OT.

So after doing a bunch of research, we ended up taking an approach that was inspired by what Figma is doing, which uses some of the same semantics of CRDTs around how you update your data structures. But we're a SaaS product, centralized server is part of the deal. So we figured we could actually rely on that to generate and guarantee some consistency without relying on sort of the distributed nature of CRDTs.

Now there's also another wrinkle here, which is collaboration really gets into every single part of your stack. The state everywhere has to be designed with collaboration in mind. And we'd already had a tech stack that we were very happy with around Apollo and GraphQL as the main state management tools. And we didn't really want to come in and rip everything out to

replace it with a new collaborative library. And we also didn't want to do a whole big single migration over to this collaboration stuff. That can be very risky from a product and engineering perspective. So we kind of wanted to be able to do this piece by piece. And this ended up being a really fascinating project, and there's actually a lot of stuff that we learned about how to work with GraphQL and things like that. But we ended up building a library on top of GraphQL and Apollo, which is our sort of main state management in the frontend that allows us to seamlessly swap out individual operations under the hood for their collaborative counterparts. And doing that we were able over the course of about two months to shift essentially our entire frontend from being single player to being collaborative with completely seamlessly to the user.

**[00:18:35] JM:** How do you test those collaborative features? That seems like a difficult feature to test.

**[00:18:41] CC:** Yeah, it is really hard. So this is actually one of the parts of our codebase that is the most well tested and we actually went and tested it pretty extensively before even touching any of the features. So we've written it kind of as an isolated library, and that library has really extensive unit testing associated with it. So one of the nice things about doing it this way is we can actually test all sorts of different combinations of edits and different orders and things like that. So we can actually be pretty robust in terms of things coming in. What order of the operation is going to come in and what's that going to do to the client state? And the other thing is we actually have a set of pretty extensive integration tests to make sure that the full stack all the way up from sort of the client libraries down to the database are working properly.

**[00:19:29] JM:** What about scalability? Has there been any challenges in building scalable products? Because some of the data sets that people are going to be working with are going to be gigantic.

**[00:19:37] CC:** Yeah. Scaling is one of the things that we focus on a lot here at Hex. At a high level, the power of our platform is also one of our greatest infrastructure challenges. And this is that we're allowing people to write code. And code gives you the ability to do a huge number

of things. So it leads to a lot of challenges around scalability. For example, a user could try to read 100 gigabytes of data into memory, and you have to be able to handle that gracefully.

Right now we're leaning a lot on Kubernetes' scaling ability uh in order to handle a lot of this stuff, but there's a huge amount of stuff down the line that we're thinking about building into our infrastructure. More intelligent sharing of resources, making some of this resource allocation even user configurable so users can know kind of – They may know ahead of time how intensive their application is or even sort of tracking and providing feedback to users about how many resources they're using and allowing that to –Using that to really figure out better how to scale the various applications.

**[00:20:42] JM:** Are there any other demands that users have put on the Hex application that have surprised you or made you have to re-architect certain facets of it?

**[00:20:52] BM:** Yeah. I think Caitlin alluded to this a little bit earlier, but one of the fun parts about building a product that is as flexible as Hex where we're effectively giving you the ability to come in and sort of write arbitrary SQL and Python is people wind up using it for a lot of interesting stuff. And so I'd say probably the most fun part of any given day for me is looking at what users have built and having them show us that. And so there've been a lot of workflows that users have sort of wanted to pull Hex toward that we may not have necessarily sort of predicted, but we've been able to support.

I mean like one example, and this sounds kind of simple, that is like users not just wanting to build applications that pull data and visualize data, but actually then wanting to have that be editable in some way and even enabling their users. So like the people who are using the applications they build to be able to do some of these write-back workflows. This was not something we had really thought to focus on early on. But as we saw what people were trying to do, it was like, "Great, we should be able to support this in a first-class way." And so those are some of the features we're actually working on now around being able to have editable tables and buttons that can do some logic control and things like that. These are the types of things where I think it's fun to sort of let our users guide us a little bit. We still have obviously a

vision and a road map and a sense of where we want to go. But in our experience the best way to build incredible products is really listening to your users and having them really show you the value that they can create if you empower them. And so that's been a really fun process so far.

**[00:22:26] JM:** You write about these different categories of data users. So you have the creators, the consumers and you kind of have the knowledge organization that sits around the creators and the consumers. And I just love to hear you talk a little bit more about the roles of the data users and how you see those different people interacting around Hex.

**[00:22:56] BM:** Yeah. That's a great question. And so I think if we kind of back up for a moment and we look at maybe a little bit more of like the secular trends here, you're seeing the rise of companies really focused on becoming data-driven and you're seeing them hiring more and more data analysts and data scientists. And there're just more of these people coming in to the industry. This is some of the most popular things to be studying in undergrad now are around data and data analysis and data science. And so I think as you see more and more of these folks coming on board and coming into companies, the question is how are these data teams creating impacts. How are they creating – A common question we actually hear from some of our customers especially folks in roles where they're leading the data teams or CTOs is what's the ROI of our data team? Like we're hiring these data scientists, we're hiring these data analysts, we're investing in new technologies like all up and down the data stack. How are these things actually creating sort of like bottom line impact for the company?

And I think you have this class of data teams and data workers that now are sort of doing a lot of interesting work and the question is how are they able to connect that and make that impactful? And so we sort of see these as our creator type users. And I think the philosophy with them is really enabling them to have very flexible tools and very flexible workflows, pull in data, work with it the way they want and then connect it to the rest of the organization and really bridge the gap between the work they're doing and what other people want to do. And so that comes into another type of persona, which are folks on other teams. You sort of think of the types of folks that might be on a marketing team or finance or leadership where they're

very data-oriented, very data-driven people, but they're also not necessarily firing up a Jupyter Notebook locally and pulling in a bunch of Python libraries and doing their work that way. And I think the question is how data teams can take the work they're doing and communicate and share it with these folks in a way that's actually usable and useful.

And then there's sort of this overall problem that you alluded to, and we've talked about a little bit around, discoverability, which is another sort of pain point you see is, "Well, great our data team, they they're building these analyses or developing these dashboards and tools," but then this stuff sort of gets scattered all over the place and it's kind of tough to find and there's this sort of like classic thing we've seen at a lot of customers. Like you have a Google drive full of slide decks and docs with pasted in charts or your Slack channels are just full of links to random one-off screenshots of charts or whatever.

And so really trying to build a product that also lends itself well to helping with organization and discoverability not just for the creators themselves, but a tool where someone remembers that the data team created a really interesting analysis a few months ago and they want to be able to quickly find it and go back and look at it or refresh it. These things should be much easier than they are today and that's something that we're thinking a lot about.

**[00:25:57] JM:** How have you seen data workflows change over the last decade?

**[00:26:01] BM:** Yeah. I think when I started my career in data, the teams I was working on and working with were primarily just using Excel. Like I started in consulting, I was basically an Excel jockey. All the keyboard shortcuts burned in my mind. But then what you've seen is over the last, I would say 10 or so years especially in the last five years, a lot of these people that might have been spending most of their time in Excel are either wanting to learn or picking up new skills around adopting languages like SQL and Python to be able to do their work instead. And these languages have a lot of advantages over some of these legacy workflows in terms of power, the logic you can build and the flexibility and scalability of the things you can do.

And I think this is where you've seen the popularity of things like notebooks. This is where you've seen the popularity of things like a lot of SQL editors. And I think having spent a lot of time with data teams and working with data teams, this is where we've really seen this gap around making work shareable. And I think there's sort of like paradox of like if I move my work from a spreadsheet to something like Python, I get a lot of advantages. But then it just becomes fundamentally less shareable and sort of less useful. I can share a spreadsheet really easily, but I can't easily share like a Python script or a python notebook I built with the rest of the business. And I think this is sort of this pain point that we've really been focused on and there's sort of a question of, "As these workflows evolve and mature and as you have users that are relying on some of these newer technologies to do their work, how do you empower them but also still have that work be really shareable and useful for the rest of the business?" And I think there's a lot of room left to explore there.

**[00:27:44] JM:** Are there any other organization-wide frictions that you see in the data analysis and data sharing world that you'd like to address over time?

**[00:27:55] BM:** That's a really good question. I think the problem we're really wanting to solve is around the sharing and collaboration, but we think that's so core. We think the ability to take work and make it into something that anyone else can consume and benefit from is really important. I'd say there's a couple other areas that are interesting to us where I'm not sure I would say we're necessarily trying to solve, but in some ways I'm almost looking for people to come along with really good answers here and things we'd love to integrate with.

I think one is around data discoverability itself. We're interested at Hex and sort of discoverability of analyses and data projects and work people have done, but I think a little further down the stack I think there's some interesting energy right now around people that are working on things like data catalogs and metadata cataloging. There're some interesting open source projects that have been spun out of some bigger companies. You have like Admunsen. That was the tool from Lyft. And there're a few others. And I think we're really keeping our eye on this level, this part of the stack and seeing what comes out there, because we think just having better organization of data, better discoverability of data and better insight into the

quality of that data is going to be really helpful for our users as well. And again, I don't think that's something we're necessarily going to bake into Hex. I don't know that that's where we'll create the value, but we're definitely thinking about how we're going to integrate with tools at that level of the stack, because I think that will make the ability to sort of, within Hex, discover data and analyze data and make sure you're working with data that's reliable and trustworthy all the easier. And that's really, really important.

**[00:29:30] JM:** Caitlin, are there any other engineering challenges that come to mind when you think about the hardest things you've had to build to support Hex?

**[00:29:41] CC:** I mean honestly there are. There are a lot of really challenging things. I think even things as simple like we're a pretty heavy-duty single-page web application that has a lot of performance implications for the frontend, stuff like that, that a lot of people I think take for granted or actually end up being really interesting engineering problems with fairly novel solutions for something like Hex. But I think, honestly, the single thing that comes to mind for me most in terms of what we're thinking about for engineering challenges right now, and in addition to the ones we've already talked about around collaboration and scalability, is security.

So when you're touching other people's data and potentially hooking up to people's cloud environments and things like that, a lot of these companies, they're data-driven companies and their data is part of what makes them valuable as a company. Sometimes it has additional restrictions on it, it's HIPAA data, or financial data or things like that. And so security as a first-class operating principle is really one of Hex's core engineering challenges I would say. And this ends up being similar honestly at its core to the scalability problem, because what we're allowing people to do is write arbitrary code.

And normally when you're trying to engineer for security, you're trying to prevent arbitrary code execution. You sanitize your input strings and things like that, but that's what that's what the product is. And so trying to architect for security from day zero has been really important to us.

And I think that's going to be one of the largest challenges we face going forward on an engineering side as well.

**[00:31:17] JM:** Is there a challenge of fitting into the existing tooling that the average company already has? So like an average company already has their workflows around data science, Jupyter Notebooks. I don't know, Spark jobs and whatnot. How do you squeeze into or how do you make the product adaptable enough to fit into all kinds of different workflows?

**[00:31:43] BM:** That's a really great question. I think what we've seen is that most companies we're working with have really well-established workflows and infrastructure. I would sort of frame it as like lower down the stack. And we touched on this a little bit earlier in terms of the evolving landscape with things like the ETL space we're using for data warehousing and things like that. I think what we're seeing at more of the layer we're operating in is a little more fragmentation to be honest. I think most companies have a BI platform that they're using, be it something like a Tableau, or a Looker, or a Metabase or a lot of the sort of interesting sort of core BI tools. But in terms of data science tooling, it's still a little up in the air. You do see a lot of people using local notebooks. You see a lot of people with very different patterns in terms of how they're managing these.

And so I think that's one of the places where we see it as an opportunity to both build a solution that works really well for a lot of people. But then as you mentioned also, integrate well with a lot of the things they're doing. And so as an example, full compatibility for us with the sort of existing file format standard around Python notebooks is really important, because we want people be able to bring their work in. We want people to be able to get their work out and have it be very compatible with different tools they might be using.

You also mentioned Spark. I think that's a really good example of there're companies that haven't adopted that technology at all. There're other companies that have basically built their entire business around it. That was one of the things that we built out over the last few months, is first-class support for Spark. And that's basically an optional thing that companies can turn

on and utilize if that's important for their workflows. Or you can completely ignore it if it's not. And there's a lot more of that we're going to build over time.

And then I'd say the other interesting thing here is just from early on as we sort of navigated the idea maze here and settled on the approach we wanted to take, really adopting and embracing and investing in the Python ecosystem, which has really exploded over the last few years especially around data science and data workflows generally, provides a lot of benefit here. You can, within Hex, just as you could in any other Python environment, effectively import whatever library or package is important to your workflow. And I think that gives us a lot of sort of built-in compatibility and integration advantage without needing to build things ourselves necessarily, right? If you want to leverage – If you want to pull data from Stripe or something, there're APIs and libraries that already make that very easy to do within the ecosystem. And so I think it gives users a lot of flexibility and it doesn't – I think we then are in a position where we're not as opinionated on how we want users to do their work. And I think that's one of the failures you see with some other – Or shortcomings you see with some other types of tools where they feel like they end up having to build everything and every integration and everything they want to do and every feature. I think sort of allowing users to always fall back to that release valve of code and be able to leverage the really rich ecosystem around this stuff is really helpful and it's great for our users to be able to do what they want to do.

**[00:34:55] JM:** Does Hex try to address the problem of data catalog? That's not something we're focused on ourselves. I think we're interested in how we can help better organize and have more discoverability around the sort of analytical artifacts themselves, whether it's just an analysis or something that's been turned into sort of an app or dashboard or tool or what have you. I think the data catalog layer is an interesting one, and there's a lot of interesting energy, and I think we're going to see sort of Cambrian explosion of tools at that level. I would say that's one of the types of things in the data ecosystem that we're really eager integrators with. And I think as companies adopt some solutions around data cataloging and metadata cataloging and even some of the things you're seeing around metrics definition and management, I think that's something where we're thinking about building really awesome integrations with because we think that that will be very useful for our users as they're looking

to leverage accurate, reliable data in the analyses that they're doing and the things they're building within Hex.

**[00:36:06] JM:** How does the Jupyter notebook experience in 5 years or 10 years compare to that which we experienced today?

**[00:36:17] BM:** I think you mentioned Jupyter Notebook, and that kind of refers to a very specific thing, a very specific open source project that we've loved. I've used Jupyter Notebooks for many years and I think it's a really vibrant open source community and there's a lot of interesting things happening there. And I think you're just going to continue to see that core open source project evolve and there'd be a good amount of innovation around that ecosystem. And we're very sort of excited and eager participants in that ecosystem.

I think there's then the separate question of like where does the notebook format go more broadly. And there's certainly a lot of other projects that you're seeing that are adopting that same notebook framework and paradigm and taking it in some interesting new directions that are separate and apart from the core Jupyter project. And that concept of notebooks I think was originally sort of popularized with even Mathematica, which I think was well before the sort of IPython project, which became Jupyter.

And so I think this is a format that makes a lot of sense for a lot of people, and I think we're going to see a lot of innovation around I'd say a couple key areas probably. One is just what we're really interested in, which is making these artifacts much more collaborative and sort of useful. I also think there's a note here of how these things can get more accessible. I think the experience for most people today that are not – I would call them like analytically technical. Maybe they're your Excel power users up into your sort of entry-level data scientists. It's actually pretty painful to use some of these things today. Like Jupyter, despite it being a really awesome tool, it's still pretty intimidating for a lot of people to sort of get started. You're needing to install a bunch of stuff locally and run a bunch of scripts and environment management is a little bit of a mess, and I think it's the type of stuff that makes a lot of sense if

you're you know spending all day in it or you come from more of a software engineering background but is not necessarily as accessible for most people.

And this is one of the things that really interests us as well is like how you can just take these workflows, strip away some of the overhead and complexity and make them much more accessible for a broader number of users. Again, I think there's this large and growing number of people that I would sort of call analytically technical who are looking to graduate out of just Excel. And I think the notebook format is one that can do a lot of good and be really useful for them. And I expect to see a lot of innovation in this space over the next five years and we're certainly excited to be part of that and contribute some of our own ideas there.

**[00:38:51] JM:** Any other predictions from you or from Caitlin as well about how engineering or data science more broadly will change in the next 5-10 years? The whole data stack?

**[00:39:06] BM:** Well, I'll offer one, which is I think today a lot of the differentiation between different roles in the data world are defined a little bit by technologies and tools. And what I mean by this is if you sort of go through your typical company that has a data team, you might see people who call themselves business analysts that are mostly working in spreadsheets or BI tools. You might be looking at data analysts, and these are people who are working in things like SQL and maybe some Python, and then data scientists who are working in you know coding languages like Python and then data engineers who are sort of doing a different subset of workflows. I think some of these lines will blur over time as some of the tools make more of these things more accessible and some of the skillsets become more widespread.

I mentioned this a little bit earlier, but when I started my career, if you were working with data, you were probably working in spreadsheets. And I think today a lot of new folks who are graduating or coming into these roles or who are already in these roles and are looking to learn new skills are picking up some of these new technologies. And it's not so clear anymore where the differentiation between sort of a data analyst and a data scientist and these things live. And so I think we'll see more of a blending of these roles and titles. I think we'll see less of the differentiation being driven by sort of the technologies and languages people are familiar with.

And I think we're going to see an explosion of people who can be really, really productive and work with data in more and more sophisticated ways over time.

**[00:40:48] CC:** I think on my end, another thing, another trend that I see starting to happen right now and I see continuing over the next you know 5, 10 years, and this is coming at kind of from a software engineering background, is over the last five years I would say you've seen this explosion in people working in Python instead of Excel and a lot of this stuff. But you come in and there're not necessarily a huge number of really well-established best practices the same way that you would have in certain fields of engineering. I remember when I first learned how to program in Java, reading a book in 2004 called *Effective Java*, and a lot of that stuff is still very true today in how you think about designing Java programs.

What I see emerging from a lot of the data science community today is techniques and strategies similar to the types of things that you would have learned for old school software engineering, but how to do data science well? How to do it as a discipline? How to make your results reliable? How to make them perform it and how to build? Sometimes it's actually just incorporating software engineering best practices into actually your data science workflow. So I think that's another trend that we'll see over the next five plus years, is right now there's this explosion of different tooling and different libraries and all of the and all of these things. And I do think that we'll start to get to a place where there are a lot of well-understood strategies around how to do – The same that which we've seen with data pipelines over the last five years, we've learned a lot about how to build stable and reliable data pipelines. I think we'll see a very similar transition with the analytical part of the stack over the next five years.

**[00:42:34] JM:** Well, it sounds great like a great place to close off. Do you guys have anything else you want to add about hacks or data or anything in general?

**[00:42:43] BM:** No. Just that like most of the companies I'm sure that you're talking with, we are hiring and we're looking for amazing engineers who are really excited about this space. So folks can find our website at [hex.tech](https://hex.tech), and we'd love to talk to folks who are really passionate about building the next generation of the aged tools.

**[00:43:00] JM:** Awesome. Well, thanks for coming the show. It's been a pleasure.

**[00:43:04] BM:** Thanks a lot, Jeff.

[END]