Vicki Boykis (Guest) (00:00):

... but I think it's a great way to work remotely. I think it's a great way to have those conversations and to document stuff that's there for reference in an async way.

Peter Wang (Host) (00:12):

You're listening to Numerically Speaking, The Anaconda Podcast. On this podcast, we'll dive into a variety of topics around data, quantitative computing, and business and entrepreneurship. We'll speak to creators of cutting-edge open-source tools, and look at their impact on research in every domain. We're excited to bring you insights about data, science, and the people that make it all happen. Whether you want to learn about AI or grow your data science career, or just better understand the numbers and the computers that shape our world, Numerically Speaking is the podcast for you. Make sure to subscribe. For more resources, please visit anaconda.com.

Peter Wang (Host) (00:45):

I'm your host. Peter Wang. This episode is brought to you by Anaconda, the world's most popular data science platform. We are committed to increasing data literacy and to providing data science technology for a better world. Anaconda is the best way to get started with, deploy, and secure Python and data science software on-prem or in the cloud. Visit anaconda.com for more information.

Peter Wang (Host) (01:07):

Hello, and welcome to the podcast. And I'd really, I'm just so glad to welcome Vicki Boykis on here with us. We're really honored to have you on the show. And while some in our audience may already know you, many are probably already familiar with you, but can you quickly share just a bit of background on who you are and what you do, for those who are not familiar? And maybe tell us a little bit more about Tumblr or about Automattic.

Vicki Boykis (Guest) (01:33):

Sure. Thank you so much for having me, Peter. I'm so excited to do this. So, as you said I'm Vicki Boykis. I am a machine learning engineer at Tumblr, which is part of Automattic. So, Tumblr itself is, as I describe it, the antisocial media social media platform. It's been around for 15 years now. Wow. And there's no one single way to describe it, but a good way would be it's a home for artists. It's a home for creators. It's a where place you can express yourself. And it's part of Automatic, which is the larger company that owns companies such as WordPress.com, Day One, and WooCommerce. And it's basically all about making the web a better place, a place to express yourself, a place where you can sell and create your content.

Peter Wang (Host) (02:15):

That's fantastic. And you called the, did you say antisocial social media?

Vicki Boykis (Guest) (02:20):

That's the way I like to describe it, antisocial social media. Yeah. So, one of the cool things about it is you don't have to use your real name. There's a lot of artists, there's a lot of creation going on. And the way our users like to think of it as kind of a also stream of consciousness.

Peter Wang (Host) (02:34):

Right. I'm sure we'll come back to this topic later, but this idea of actually having a free space to be who you are or to express yourself may actually make, it's a different kind of socialization. And one might say the amplification platforms that we're familiar with today, the ones on which you and I are so famous, perhaps, but still, those are actually more antisocial in certain regards. They certainly incentivize a lot of antisocial behavior, unfortunately. But we'll get into all that later. But maybe to start off with, I will just seize on something you said. You call yourself a machine learning engineer. And did you always call yourself that? Or at some point, didn't you call yourself a data scientist at some point, or was it always ML engineer?

Vicki Boykis (Guest) (03:11):

Yes. It's been an interesting evolution. My background is actually in economics, which I think is true for a lot of people in the data industry. Probably the two most common jobs this year are economics and physicists. We love to congregate in data. And I actually started as a data analyst, and kept running up against this thing where I had to do more and more software development. And I found that at first, I wasn't too interested in it, but then I basically saw what software development could do. So, I think the first job I had, I was basically an economic analyst. So, I actually graduated with a job in my major, which I think is pretty rare.

Vicki Boykis (Guest) (03:45):

And I had to analyze some data from the World Bank, and it was on the World Bank website. And I was taking a row, and it was in like this horrendous columnar format that you see in all open-source data everywhere. So, I would just take a row at a time, highlight it, copy it, and paste it into my Excel sheet. And then someone showed me at work, they're like, "Did you know that you could scrape this data with a Python script?" And I'm like, "What?" So, once I saw that you could do that, I thought I'm all in on what this entails. And then I went on to do more and more progressively programmy things. So, I did go into data science then after data analysis.

Vicki Boykis (Guest) (04:22):

And then at one of my jobs, I had the fortune or unfortune to be placed on a team that was being introduced to Hadoop at the time, as a data scientist. And I started writing, it was at the time pig scripts. So, there's this meme going around Twitter, tell me how old you are without telling me how old you are. So, one of the things I did was pig scripts, which was the original data science. And then I thought, "This is really cool. How can I do more with this? How can I get more involved with this?" So, I moved progressively more and more towards looking at the data side of things to actually creating and programming these systems.

Vicki Boykis (Guest) (04:56):

And I think in a way that's where the entirety of data is headed. Not that everybody is writing code or fortunately interacting with Hadoop, but it is getting to a point where, I think we started in data science where everybody was doing these analyses, these one-off analyses Excel, R, et cetera, just on their machines. And they got to the point where, "Oh, okay. This is a really cool thing. We can make it part of a product. How can we actually build this into our application? How can we build recommendations into our application? How can we build these machine learning models?" So, I think everybody is getting to the point where we're moving closer and closer to what we consider to be software development and

production environments. So, I think my career progression has, hopefully not just like a one-off bias, but has reflected where the industry is headed as a whole.

Peter Wang (Host) (05:43):

Right. Well, and what's interesting about that is that we look at this and we say, well, as data science goes from exploration to producing things that are part of production, production applications and whatnot, data science has to mature in certain ways to meet the standards of software engineering and what people who operate these systems and get paged at like 2:00 in the morning, like what they would expect from a certain level of quality and whatnot. And that all makes sense. But in life, everything changes everything else. So, what are the ways that the requirements of these data systems, these online prediction systems, different systems, what are ways that you're seeing these change how software engineers in traditional application development, who are used to writing CRUD apps or little database things, or little whatever things, as they start looking at these giant data-driven things and deploying them? How does IT and software, how does it change or how have you seen it change, or maybe reluctantly, or try to refuse to change? Tell me about that part of the interaction.

Vicki Boykis (Guest) (06:40):

I think there's a really interesting interplay between what we consider traditional software development, and what we now see in machine learning or data-driven development as a whole. So, like you said, so the traditional development life cycle is what? So, you have a CRUD app or you're developing a feature, let's say you're developing a toggle to do X, Y, and Z in your app. So, what do you do? So, you develop this toggle, you test it, you have unit tests, does this toggle work on and off, et cetera. You launch it, you put it behind a feature flag, and that's it. You're done with creating a feature.

Vicki Boykis (Guest) (07:12):

It can be pretty open ended in some ways, is this the right framework to use? Do I need to architect this toggle to be able to be hit a million times a second, whatever. But in general, it's pretty close ended. And the way that data development works, it's very open ended. It's very different because how does machine learning work? So, you have some data, you have to spend a generous amount of time, 80 low ballpark estimate, cleaning it, getting it ready for features, getting it ready to be part of a model. Then you create the first iteration of your model. And that might be a good model, or it might be a bad model. What does it mean for it to be a good or a bad model? Maybe it could be classifying things correctly, or it could be classifying things incorrectly, or it could be serving you recommendations. What does it mean for a recommendation to be good or bad? It could be good for you, it could be bad for the platform, et cetera. It could be good in general, it could be bad in general as well.

Vicki Boykis (Guest) (08:04):

So, what does it mean for the results of a model to be good or bad? Obviously there has to be human-in-the-loop evaluation on that. And then do you do a second pass at this model? So, maybe you do, then you create more results from this model. Is this a good model? Is this a better model than the first model? So, if we do have what we consider to be a bad model, for whatever metrics we follow, whether they be offline metrics on model fit, precision recall accuracy, or online metrics, are people clicking on the results of this model, we might need to go back and collect more data. Or it might be that this works, and then the data that we're collecting changes, someone starts logging something

differently. All of a sudden we have null values for certain features. So, then the model doesn't work anymore.

Vicki Boykis (Guest) (08:49):

So, the machine learning model, the process of creating a machine learning product is a lot more open ended. And I think that where we bump up into the traditional constraints of software, is software says this works, or it does not, or you get an exception. That's it. With machine learning, you also get exceptions. You get a lot of exceptions, a lot of YAML validation errors, but you also have this thing where there's a lot more human judgment that you put into the system. And I think that we're just starting to figure out how to make space for that in a development cycle.

Peter Wang (Host) (09:20):

Right. Now, that all makes sense to me. I mean, the way I see it, a traditional software developer has the luxury of being able to specify what correctness is, in general. You start knowing what correctness is. I mean, you spend a lot of time trying to get the business requirements nailed down. So, that does take time. But once that has been moved semantically from the business domain into some kind of specification or something like that, a technical specification, you can then build systems that then have inputs, but you know if it's right or wrong.

Peter Wang (Host) (09:50):

But with data, I call them value-dependent systems, that even just, you imagine when you give a coding test to someone like, "Hey, write a function that reverses a legal list, or you write a function that adds a list of numbers or something," and there's a correctness aspect to it that once you write that function, the function is correct regardless of the values in the list, sort of. But then all the data people know, no, actually you can write a function and it's right for some values and wrong for other values, and holy crap, how do we build an entire production system where every piece has this kind of a wiggle to it? And that is incredibly difficult.

Peter Wang (Host) (10:22):

And I think, I'm going to go on a limb here, but I suspect also that this intrinsic complexity of building ML systems and deploying ML systems is obscured sometimes by a cultural impedance mismatch between the kinds of people who are often building these systems, versus traditional people who are building database applications and business software app kind of things. Because you're right, they're economists, they're like washed up physicists. They're people who come from this data world, and they end up being data or ML app builders.

Peter Wang (Host) (10:53):

So, at least where I found doing consulting 10 years ago, there was definitely a bit of, "You're not from around here. Are you?" experience when you met up with a whole room full of Java architects. You're like, "No, I haven't read all the books, or I can't speak all the Java architect lingo." But I'm telling you, this is a thing you have to solve. And this is actually really hard. Not because I'm an idiot, but because it's really hard. All these computer systems are the result of humans interacting with each other, and

humans, tribes of humans form tribal identities and all sorts of baggage that makes this difficult. Have you seen that in your experience? Do you see any flavor of that?

Vicki Boykis (Guest) (11:27):

Yeah. I think something that you touched on the human aspect, I think it can be sometimes hard to reconcile the fact that all of the data that we're working with to put on into these systems is human-generated data, for the most part. Unless you're dealing with, I don't know, statistical control and factories, et cetera, but even then humans play a part of that process because humans built those factories. So, there's a shape to the data, there's a flavor, there's a nuance. There's a way to understand this that is inherently, like you said, at odds with traditional software development, which is we have a functional spec, we put data in, we get something out. We don't really care about what we put in and what we get out. And if you notice that it's true, like when you do unit testing for traditional software, we only care that this function works, and it's not a focus. Or you're not supposed to have a lot of data that you mock out because the idea is just the function works or it doesn't.

Peter Wang (Host) (12:21):

Well, you're supposed to test extreme values or certain kind of error conditions, right?

Vicki Boykis (Guest) (12:25):

Yes. Right. But you're not supposed to test like you have potential values for this, what happens with this, what happens with this. But you do need to pay more attention to that in machine learning as well. And I think these two different mentalities, it's something that I've thought about a lot. When we're doing software engineering, we care about building things that data can just pass through easily. When we're doing machine learning, we care about the data itself just as much. And I think those two things both compliment and are at odds with each other.

Peter Wang (Host) (12:55):

So, there's absolutely, there's this amazing quote from Jim Gray that I'm going to try to find here. I just presented this, well, I've been presenting this for a long time because you can make yourself sound really smart by quoting smart people. It's a cool trick. It's a cool conference talk trick. But Jim Gray has this really great quote, and he said, what is it? "The separation of data and programs is artificial. One cannot see the data without using a program. And most programs are data driven. So, it is paradoxical that the data management community has worked for 40 years to achieve something called data independence, a clear separation of programs from data."

Peter Wang (Host) (13:32):

And when I read that however many years ago, it sort of blew my mind in the sense of like, oh yeah, no, you're right. We build database systems, or Hadoop, pig scripts, whatever it is. Or a distributed Oracle, standard you know, standard SQL database, but we have a data management program that does data management that's independent of the data.

Peter Wang (Host) (13:51):

And there is some regime where software that is data independent is still useful. It's useful because it's better than pen and paper. It's better than people calling each other up. So, it's a little bit useful in that

regard. But now we're going into a world where we're realizing, oh, data-sensitive, value-dependent computing, programs that are the synthesis of code and data are incredibly powerful, undeniably impactful. But we have absolutely no theory as an industry to manage this really. We're just starting to figure this out. And I think you used this quote before where you said that we're still in the steam-powered days of machine learning. What do you think will demarcate the end of the steam power era in the beginning of the fossil fuel era or something like that or, I don't know. What is the post-steam, I guess that was also fossil fuels, nuclear-powered ML?

Vicki Boykis (Guest) (14:37):

Yeah. I think we're getting to a point, so just to back up and give a little bit of context to the history. So, we had this era where we had compute that was close to storage in a traditional database system. Then we took those traditional database systems apart because storage became very cheap, and transferring data over the network also became cheap. So, it was easy to separate them and create the HDFS data lake, et cetera, S3 era. You just throw everything that you have in there and kind of let Spark sort it out. So, that was the past, I don't even know how time moves in the data science machine learning life cycle anymore. So, that was the past 10 years. Now we're getting to the point where people are like, okay, we have these systems separately, but now we want to keep them close together. So, now we're seeing the bundling, like who was it that said-

Peter Wang (Host) (15:26):

Yeah, all business is bundling or unbundling. Right.

Vicki Boykis (Guest) (15:29):

Everything is either you make money in either bundling or bundling right. And the same with data too. So, we started bundling it, then we unbundled everything. We put it in MongoDB or we put it in HDFS and then we just had our program sort it out. And now we're getting to the point where, oh, we have a lot of stuff that's unbundled all over the place. Now we want to keep a close eye on it. So, now you have technologies like Delta Lake, for example. Postgres just released a machine learning module where you can do machine learning in the database itself. DuckDB is a way to move the database to machine learning. So, there's different theories on how this is happening, but I see all of this as moving closer together.

Vicki Boykis (Guest) (16:08):

And then there's also a larger amount of architecture going on around these systems. So, MLOps, obviously everyone's talking about MLOps, everyone's doing MLOps. So, now you have MLOps, which is like, okay, we had these systems in the wild, maybe we want to now manage them like we manage actual software and write actual software, and you have monitoring and you have this and this. And I don't think we still know entirely how to do it, but we're working to the point where, again, back to what I was talking about with where my career's headed, we're working to the point where we're getting closer and closer to software engineering. We don't know exactly how it'll be different yet, but we're starting to congeal with data and compute together.

Peter Wang (Host) (16:46):

Well, so because this podcast is called Numerically Speaking, we can be a little bit more opinionated. And as an ex-physicist, we can have the physicist conceit we can always bring to the table and say the

software engineers don't know what they're doing either. Because they will, the senior software architects will be the first to tell you software is architecture before the arch. It's still, it's also one of my favorite quotes about software, is that science advances by scientists stand on each other's shoulders, software developers move forward by standing on each other's toes. And that's essentially layers and layers of bandages that basically is like sedimentary compress into becoming substrate. And because the poor hardware engineers work late into the night to make things faster, they spend all this time making things a little bit faster, and then we just throw like another VM abstraction on top of it, because why not? Because we can't be bothered read the docs.

Peter Wang (Host) (17:29):

So, I feel like even as we're saying we want to make this stuff more and more robust, we want to figure out how to productionize it, I don't think that the way people productionize software should be the gold standard. I don't think it's very great on that side of the world either. So, maybe we can find our golden path and middle path between these different worlds. But I think certainly there are some just intrinsic hard constraints that the data world brings into this. And this is why personally, I like to try to use the word cybernetic more often, because these are control systems. These are sensing systems, prediction systems, control systems. And as we deploy ML to the edge more and more, we'll find that the sensor plus the sensor training action, sort of the OODA loop in the edge is going to become a bigger part of people's architectures.

Peter Wang (Host) (18:10):

Especially as we have data privacy, if we go to federated learning and things like that, you're not going to be able to pull it all together to some gigantic data center. You're going to have to figure out actually new application and prediction architectures that work when they're federated at the edge. So, I really want to make sure for people who are listening, that we hold onto our own set of values around this, those of us who are mathematically inclined.

Peter Wang (Host) (18:31):

Well, the other thing I wanted to ask you about around some of this stuff, so the high level, this consolidation of compute and storage and the emergence of MLOps, of course there's a lot of conversation around the modern data stack, data Twitter threads about some of this stuff, mocking it, talking about it seriously. Where do you think, why is there such a Cambrian explosion of not even just projects and companies, but even entire categories right now? It seems like there's more now than there was before. What is it? COVID? Is it more people showing up on Twitter and it passed some threshold? What is it?

Vicki Boykis (Guest) (19:00):

When you're talking about categories, you're talking about what the modern data stack is, or just generally these classifications or what we think about?

Peter Wang (Host) (19:10):

Yeah. So, the modern data stack set of conversations are around... I mean, that sits a little more squarely in data engineering, the intersection of data engineering and data science, let's say. But then MLOps and all the Kubeflow and all the other kinds of things, that's in a kind of a different space. That's the intersection of data science and more of like IT and software development and software deployment.

Peter Wang (Host) (19:29):

I actually saw the word deep ops. So, for operationalizing deep learning. Don't laugh, it's not nice. We don't laugh at people when they use terms like this. We try to understand where that's coming from. Because actually, most of the MLOps stuff is container based, and that's notoriously difficult to do GPUs very well with those things. So, do we need different IaaS providers to do deep learning operations or is it different? So, all these startups are coming in, VCs throwing money at them. New terms are being minted faster than Gartner can even make quadrants. And at the end of the day, where does someone even get started? How do they even know what is good and what is crap or what is hype? Give us some clues here.

Vicki Boykis (Guest) (20:03):

Yeah. I think not a lot of it, but some of it is partially due to the way we do compute now, which is in the cloud. So, the way we used to do compute and software development and data analysis was that, so for data analysis we would do locally, software development we would have a monolith. You would have a big monolith, you would work on that monolith together. Now everything is being broken up. So, now what do you do if you have a monolith, you have to break it up into microservices for some definition of have to. So, now what we have is we have a lot of, and what are microservices? So, if we're in the cloud, what it means is you have this service that does this. You have this service. You're basically gluing different things together.

Vicki Boykis (Guest) (20:43):

So, we had the monolith, we broke it apart into microservices, now what do we have? We have 10 different services for machine learning that you have in AWS or GCP. And you're just basically trying to get them to talk to each other. So, I think a lot of this comes out of the fact that you now need a holistic view into these systems. And like you said, machine learning systems are very, very complicated, maybe even more complicated than software systems because of all the nuances that you mentioned. And especially because we need to have humans in the loop at some point for a lot of these systems, evaluation, data-specific privacy, GDPR, all of that stuff, federated privacy, all of that stuff.

Vicki Boykis (Guest) (21:19):

So, what happens is people are, I think a lot of this comes out of the fact that people are trying to wrap their mind around this whole system in some way, and try to control and reason their way through this whole system, as it applies to their entire stack. And that's where I think a lot of these startups come in, that's where this management. We have a lot of layers. Software development today, you might be working with, I don't know, 10 different layers, context switching between three or four different languages at a time. So, I think this is a way to try to get a handle on all of that.

Peter Wang (Host) (21:50):

The way I think about it is that a lot of people end up, they could be very tenured in a particular area of software development and the world's changed around them. There's just like 10 different new things they have to consider. Or they could be relatively junior, and they haven't had, like me, 30 years to understand the fundamentals of computing from 8086 up. So, they're just trying to figure out the Linux bits here, the Kubernetes bits there, front end bits here, some linear algebra there. It's a lot to bake and understand. So, coming from a point of empathy, when people just are getting hurt by all these different

things, they're stumbling in the dark on tables they can't even name, they're stumbling their toes on furniture they can't see, so they just give names to baskets of pain.

Peter Wang (Host) (22:31):

It's like, "Okay. This part of the room, I call some..." And they're always called ops. If we actually know what we're doing we call it engineering. So, this basket of pain, we call something ops. This other basket of pain is some other ops. And then we just got to do. And every now and then you just torch the whole thing with blowing away your MPM modules, [inaudible] modules. Anyway, so one thing when we were chatting before about some topics, one thing that came up was I reflected that I really enjoyed reading your newsletters in your Substack and the normcore stuff, because you took a very candid, I would say, a very sort of sensible like, "Look, I don't think you need all this stuff. And here's what I do, and this works."

Peter Wang (Host) (23:05):

And I find that in as this industry is, as more people are coming into this part of the industry or as machine learning absorbs more and more things, there's a lot more posturing. There's a lot more people coming in, taking social media, trying to use social media as a way to build their careers or to hype their startup, hype some tech. Certainly for me, very disturbing is this creating GitHub, like vanity GitHub repos, or doing some activity in the open-source space, not out of a genuine desire to create, but because it's padding your career or padding your resume.

Peter Wang (Host) (23:33):

So, one of the things we talked about here is the systems of the system of incentives around community, social media, whatever around the communications that we would use in the industry in the past, that the system of incentives and the commercial environment, has that distorted the human ecology around open source, open data science and all these things? And I think it has in the last 10 years. That's my view is something has definitely shifted. It's hard to put my finger on exactly what. But you, not only just as someone in the industry, but also someone who works at a social media company, what are your perspectives on that? Is it distorting? Have these technologies and tools been distorting? And if so, what are things that we might be able to do to ground things again in reality and credibility?

Vicki Boykis (Guest) (24:16):

So, I think there's a couple things going on around social media or just around how we talk about tech and social media. One of those trends I've noticed is context collapse has gotten very, very big. So, context collapse, there's a really great book about it called *The Presentation of Self in Everyday Life*, by Goffman, basically says that individual people basically segregate their audiences based on where they are at any given time. So, in the physical world, we're able to do that really well. You would talk to your parents one way, you would talk to your children a different way, you talk to your friends a certain way, and usually, or you talk to your boss a different way from that. And usually all of those things are different representations of yourself. And something else that's interesting is, I read that basically your personality is reflection of who you're talking to at any given time. You don't have one single personality. It depends a lot on what you're talking about.

Vicki Boykis (Guest) (25:06):

In the digital world, we can't do that anymore because on very, very public social media like Twitter, for example, you're talking to everybody at the same time. So, whatever you say, if you're talking about, for example, open source to one audience, another audience might interpret it completely differently. So, we have that sense of context collapse, which then leads people to either try to be louder than the context and make very large claims that generate a lot of controversy or noise, or they cause people to be very bland and not [inaudible] any sort of controversy at all. And just basically do things like you said, where they create repos or they have threads, like here's how you do X and Y and Z, and the point is just to get followers to see that you know how to do X and Y and Z without any opinion on it whatsoever.

Vicki Boykis (Guest) (25:54):

And a lot of stuff online right now is tied to your real identity, like Twitter. Sometimes there's a lot tied to your real identity and a lot of people do use their real identity. So, that means the stakes are very high to be also very performative. And what that's resulted in, I think is a lot of performative signaling as well. So, I just saw, I was reading this story, I think it was yesterday, saw it surface on Hacker News, where basically this woman felt dehumanized by a viral TikTok that was filmed without her consent. So, it was just this guy that came up to her and gave her some flowers to hold. He said that he needed to fix his hair or something for a date. And then he walked away and he said these flowers are for you. And this thing was being filmed the whole time and she didn't know it.

Vicki Boykis (Guest) (26:34):

And she was shocked, but it went super viral because of how nice the guy was being. She said, "I feel taken advantage of because this guy was acting not out of genuine good faith, but in a performative nature." So I think there's a lot of element of that in whatever we do online, we think about how, not only what we say, but how it will play, in a way that you noted is much more prevalent now even than 10 years ago, when people were blogging about stuff. There wasn't as much of a potential for it to get out and explode. And I think people were a lot more genuine in their opinions.

Peter Wang (Host) (27:07):

So, this is a topic I could talk about literally for hours, maybe days, because I think somehow we engineered and built a bunch of stuff that jammed itself in as infrastructure for human-to-human communications. And none of the people building any of the stuff had any reading, they'd not done any of the reading or the background, on any of the stuff. And it was so easy. I mean, you build a webpage, you build a message board, people get on it and they share funny pictures and they make jokes about things. And what's the harm in that? And it's sort of like giving kids a bunch of uranium or whatever. It's like they're just playing around with this really pretty blue metal, but you put a lot of it together and the whole thing gets vaporized.

Peter Wang (Host) (27:44):

So, there's something around this, the way you talk about context collapse and the presentation of self, the phrase that I like to use is that every conversation is a space. And every humane space has consistent and implicit or explicit, but norms, has norms that are understood by the participants. And the problem with these online environments and communications media, so to speak, is that we don't create, there's no way to create a space, or the spaces shift so much. One thread here, another thread there, the norms… So, what identity, what avatar you present is completely, you have no idea. And that's deeply stressful when we don't know who we should be, how we should show up. That's incredibly stressful.

Are you talking to the CEO of Megacorp, or are you talking to social media intern who's just upvoting your little joke you made? So, there's this kind of thing that's really, calling it inhumane is maybe very harsh, but it's just something deeply inconsiderate about how this is done.

Peter Wang (Host) (28:38):

In your example about the video going viral, I've seen these other things where people they do pranks, and they've always been pranks as long... I mean, I remember TV shows where they would go and prank somebody or punk somebody. That's like, okay, fine. The social media stuff, there's one where, I guess it was three women at a gym, and two of the women were just really being nasty to this third woman. And they were basically trying to troll this guy at the gym into, and he's just a random dude, trying to tell this woman like don't listen to them. They're just, I don't know why they're being so nasty to you and all this stuff, but they were obviously trying to prank him into doing something that would then make him the fool. And it's a bullying of a sort. What you're doing is just creating an environment, a space, and that conversation, that interaction, that space is a trap. And it's entirely designed to exploit the ignorance of the other person. They're not aware they're on stage. That is a kind of bullying. It's absolutely taking advantage.

Peter Wang (Host) (29:27):

But anyway, this went a little bit off... I get so passionate about this topic because it's so obvious to me. It's like, "No, let's not give children plutonium and uranium to play with because this is actually really toxic." We need to train them to have genuine conversations. How to actually, because if you're... I guess this is the thing that it does come down to, if you feel threatened, then you're not going to present vulnerability. You're not going to approach other people with openness. And this is then a race to the bottom, then others will not respond in kind. So, overall what sucks out of this thing, what just drains away, is any kind of human-to-human, vulnerable, deep listening, actual interaction that I guess integrity. Is there a way to engineer these systems to select for genuine integrity?

Vicki Boykis (Guest) (30:09):

Yeah, that's tough. I think, I don't know. I'm only a machine learning engineer. I think that there's a lot of social sciences that need to be built into these systems that we are just now starting to understand how to build them in correctly. I think that, so I was trying to find this while you were talking about messaging and message boards, but I read a really good book a couple years ago. I'll have to find the link. Basically it talked about the first moderator of, there was an online chat community that started in New York that was specifically based in New York. And she moderated for something like, I want to say 15 years or so. And basically it was an interview with her and she was one of the first people who was a, what we consider today a moderator. And she talked about the rules for her community. She really thought this through. This was in the early '90s even.

Vicki Boykis (Guest) (30:58):

So, it was before any of this where we had to think through what content we put on a platform, what we don't. How to be nice to people on the platform, how to kick off offenders, how to all the stuff that we now are talking about at a large scale, it starts with this very human thing of how do you moderate a people in a community to be nice to each other.

Vicki Boykis (Guest) (31:18):

And I think that in smaller communities, this is much easier. This goes back to a post I wrote, “Good Things Don't Scale,” which was about the country of Iceland, why Iceland is so awesome because it's tiny, and they can make things awesome. But in a large country, it's harder because you have a lot of networks, a lot of things at play, a lot of counteractive interests. So, what does that mean? So, I think what we're seeing is we have these very large social networks, but then we have clusters. And the clusters in the social networks are easier to manage. And I think that's why we start to see things like data Twitter or modern data stack Twitter. And people can discuss those things in their interest groups, and I think that's one of the things that makes it easier to have those civil discussions, because then you start to fold in a little bit of that context collapse.

Peter Wang (Host) (32:06):

Right. Right. You fold in the context collapse, and then you make it easier to have the norms. If you have a technical part of Twitter, then it's like, "Hey, here we're talking the tech stuff, let's keep the politics out of it, keep the religion out of it, because we're here to talk about some tech." That's not to say that humans aren't political animals, that's not to say that people shouldn't manifest their spiritual beliefs and live in accord to their personal spiritual values, but just this part of the conversation, let's just really keep it about the tech or keep it about the topic at hand. And the moderation thing, that is a way of enforcing the norms. What do people get their wrists slapped for? What do people get ejected from the room for? Setting those norms.

Peter Wang (Host) (32:42):

And the thing is when we look at, well, all the discussion around free speech on Twitter and who gets banned and who doesn't, and this and that and the other, boosting versus de-boosting versus all these things, it's an attempt to use formal systems and computers to enforce a deeply human thing, which is what are the norms for X group of people. N choose K, what are the norms for K versus the norms for N? There's no formal way. One another great quote, one of my favorite, most quotable computer scientists, Alan Perlis, and he famously said, "You cannot move from the informal to the formal via formal means." So, human-to-human norms and the communications like what's accepted and how do we push the boundaries together, that's a thing that a bunch of people have to decide, and there's not an algorithm for that. This is my personal view on this.

Peter Wang (Host) (33:24):

But there's also, I would reference to readers or listeners who are interested, to listen to a great podcast that on the Jim Rutt show, he did with the creator of Slashdot. And they talked about the moderation system on Slashdot. It was a community moderation, but they handed out mod points to a few people. And when you got mod points, you only got them every now and then, and it was a special thing that you would use to enforce and upvote/downvote certain things. It wasn't like here where everyone gets to upvote and downvote, and you have mobs like suppressing things or boosting things. So, anyone who's interested in this topic, I'd recommend those two things as well.

Peter Wang (Host) (33:55):

I'm looking at the clock here, we are running a little bit out of time. I would love to talk to you about so many more things. But on the flip side of all this online stuff, tell me, we talked about in-person versus remote work. And you had some thoughts there on how Automattic as a distributed company and

remote-first company manages this. And I would love to hear from you about that and how you all go about doing that.

Vicki Boykis (Guest) (34:16):

Yeah. So, Automattic is unique in that it's been remote and distributed from the very beginning. We've been doing it before it was cool. And one of the ways that we use to manage it is we have a top-down culture and bottom-up culture of written communication first. So, I've written about this before, but the way we communicate is this product called P2, which is, as you would imagine, a WordPress blog. And basically every team in the company has a P2. You can see everybody's P2s. And literally we don't send emails, everything that is discussion, I'm thinking about doing this project, an RFC, a retro, et cetera, all of that happens on P2s. People can comment on it, et cetera.

Vicki Boykis (Guest) (34:54):

So, what that does is two things. One, documentation, it codifies everything that you're working on. I can see all the projects that I've done, and I'll just send you a link. "Oh, here's a P2 to this. Here's a P2 to that." Second, when you're working internationally, so Automattic is distributed internationally, you might have people who are 12 hours behind you, 12 hours ahead of you, people who are working six-hour time differences. So, Slack, you can use for that, and you can use Twitter threads. But it kind of slows down the conversation deliberately in certain ways so that everybody has a chance to respond at any given time because the response is a comment. So, you make a comment, and that drives a conversation in the same way that a meeting might.

Vicki Boykis (Guest) (35:35):

So, I found this, I love written culture. I love P2 culture. What is really interesting about it to me is that you can very clearly see what you've worked on, and you have references to every single thing you've done. And you can also search through every P2 so you can clearly see what a project, where it came to. Now, of course it might take you, the downside is it might take you awhile to get to what you need, because a project thread might have, I don't know, 15, 20 different comments. What if we do this? What if we do this? Here's the outcome. So, it does take deliberate filtering. But I think it's a great way to work remotely. I think it's a great way to have those conversations and to document stuff that's there for reference in an async way.

Peter Wang (Host) (36:12):

Yeah. So, I'm very envious of that concept of having such a written culture. Certainly we hear about Amazon having such a culture. Internally at Anaconda of course people write a lot of docs and there's collaborative editing of docs and there's a lot of these things. And we have we call APEs, so like product enhancements, things like that. So, there's these things that we are working on, but I looked at the P2 site, so it's wordpress.com/p2, for anyone who's interested, really looks super interesting. And I would love to actually maybe try prototyping that for some of our stuff internally. But that having that written culture, it does slow things down. And that slow down is actually, it makes things a little more deliberate in that way. So, that's excellent. So, with that, I guess one final question for you, Vicki, which is, do you have any takes on AI and the singularity? How far out you think we are? Are you building one? Have you guys produced one yet?

Vicki Boykis (Guest) (36:59):

I'm just trying to get my-

Peter Wang (Host) (37:00):

I heard that a Google language model just became sentient. So, has any of your stuff become sentient?

Vicki Boykis (Guest) (37:05):

Listen, Peter, I'm just trying to get my YAML to link correctly out here. It's hard enough to build these systems as it is, like make sure that you have your feature store, make sure you have this, make sure you have your model, make sure you have your output, make sure you put it in production. I am not worried about the singularity anytime soon. I think we have a very, very long way to go.

Peter Wang (Host) (37:25):

All right. Well, that's a very bold claim to make. I think the old joke was that I'll replace you at some point with a small Python script or a small Bash script. And I could say now, we might be looking at a Skynet creating a sentient robot just to parse your YAML's robustly.

Vicki Boykis (Guest) (37:41):

Oh, I hope so.

Peter Wang (Host) (37:41):

That would be hilarious.

Vicki Boykis (Guest) (37:47):

If that's the case, I'm all for it. I'm all for it.

Peter Wang (Host) (37:47):

Well, thank you so much for joining us. I really appreciate chatting with you. Appreciate the insights and perspectives, Vicki. And I will see you in whatever our part of Twitter. Look forward to more conversations in the future. Thank you so much for joining us. Really appreciate it.

Vicki Boykis (Guest) (38:00):

Thank you. It was my pleasure. Thank you so much for having me.

Peter Wang (Host) (38:00):

Thank you. And for the listeners, we'll have links and references to the various resources and things mentioned today. Vicki, if you figure out that, the thing about that moderator of the message board, definitely shoot that over so we'll include that. Yeah. Thanks everyone for listening, and look forward for our next episode. Thank you for listening. And we hope you found this episode valuable. If you enjoyed the show, please leave us a five star review. You can find more information and resources at anaconda.com. This episode is brought to you by Anaconda, the world's most popular data science platform. We are committed to increasing data literacy and providing data science technology for a better world. Anaconda is the best way to get started with, deploy, and secure Python and data science software on-prem or in the cloud visit anaconda.com for more information.