James Cham (00:00):

But I bet you there are a whole set of things that we do that really could be boiled down, and digitized, and structured in the same way that you'd structure something for Mechanical Turk.

Peter Wang (00:13):

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.

Peter Wang (00:42):

Make sure to subscribe. For more resources, please visit anaconda.com. 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 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. Welcome James Cham. Hi, thank you so much for coming on today.

James Cham (01:12):

Thanks Peter.

Peter Wang (01:13):

Really great to have you on the show, and I'm really excited for our conversation today. I'm sure many in our audience are already familiar with you. For those who are not, can you share a bit of background on who you are, and what you do?

James Cham (01:27):

Sure. I'm a seed stage investor in a bunch of companies. I work at a firm called Bloomberg Beta, where Bloomberg is our only investor. And I primarily invest in the future of work, which is everything from machine learning infrastructure, on up to productivity apps. And I've had the good fortune of having many interesting conversations with Peter late at night, either in person or over Twitter. I'm glad to get a chance to have one of these recorded for posterity.

Peter Wang (01:57):

I'm so excited, and I always enjoy our conversations so much. Do you want to tell us about a couple of the portfolio companies that you've invested in, that you're excited about or the kinds of things they do, so people get a sense of the variety of investments you all do there at Bloomberg Beta?

James Cham (02:10):

I mean throughout my career, I've been lucky on a bunch of seed investments. It's everything from seed investments in companies like Twilio and Dropcam, all the way up to more recently, investments in companies like Weights & Biases and Streamlit, right? You get a sense there of the things that I'm most

interested in. And to be honest, I'm animated by that combination of the founder and the idea with just the little twist that makes sense, or sometimes the inversion on commonly held wisdom, or slightly askew angle that might make a big difference.

Peter Wang (02:47):

Yeah, there's got to be an angle. What I've noticed in speaking with a lot of investors and venture capital folks is that the opportunities, a lot of people see the different opportunities, but a lot of the what I think of the great investors, they're looking for the founders, looking for the people who have an insight, a different angle a little bit on the stuff. Congratulations by the way. I saw you guys just closed a new fund. You'll tell me about that and about the kinds of businesses and founders you're looking for to invest in with that fund.

James Cham (03:15):

We just announced our fourth fund. We've got two parts of that. One is our core business which is investing about a million bucks in seed stage, or pre-seed-stage companies. The labels change, but the nature of the relationship and the entry point remains the same, whatever you call it. And then maybe we'll have a metaphor other than agriculture. And then we've also got a follow-on fund that's for opportunistically investing in companies we're already invested in. So, that's the core of it. And then the kinds of things I'm looking for are a wide range of attempts to take what I'd say are like, or at least the way that I'd put it is that we're still in the middle of figuring out what software's good for, right?

James Cham (04:00):

And whether that software's data-driven stuff or machine learning. And that connection, we're still looking for those little unlocks. And in my world, so much of that really is around unlocking developers, unlocking analysts, unlocking knowledge workers. That's the rough way that I think about it, and that's the rough place that I hunt.

Peter Wang (04:22):

We're going to get back to that. That's a very provocative...

James Cham (04:24):

And you can fold in all the buzzwords, right? Whether you want to fold in low-code, no-code, whether you want to fold in web assembly, all those words you can fold in, but that's the core part of it, right? Can we actually radically unlock productivity? And productivity is a bad word. Productivity not in the economic sense of the word, but productivity in the broader sense of actually getting people to do things, right? So, that's the mission, that's the dream.

Peter Wang (04:48):

Okay. Just on the last few sentences, we could do a lot because it's a very provocative statement to say, "We're trying to figure out what software is good for, and there are software companies with market caps in the tens of of billions of dollars." We're looking at this and we're like, "Okay, certain investors of course famously have said that software's eating the world." By asking the question, we're trying to figure out what is software good for, compare and contrast the point of view there, right? If you're coming from this point of view of... Because it sounds like you have a much more critical, or maybe you

have higher expectations for what software really could do for people, versus merely, "Oh yeah, it's better than pen and paper."

James Cham (05:29):

I mean I don't want to get too metaphysical about it, but it takes time, right? It doesn't take one generation. It takes multiple generations to really fold in some new fundamental technology, and you do have to remember that software is basically magic. The fact that we take these very big numbers, we agree on how to deal with these big numbers and break them up into instructions is a crazy idea. And the fact that we have consistent ways of moving these numbers around and then convincing people to do things based on it is I don't know. I'm conventionally religious. I'm a Protestant Christian, right? I'm super normal.

James Cham (06:04):

But go back a thousand years, the sort of things we do, if you really explained what software did to someone a thousand years ago, they really would think it's magic because it is. It's amazing that we've managed to coordinate and all agree on these things. And we've managed not just at a human level, but even all the way down to electrons, the fact that we've all agreed on is maybe one of the most impressive acts of collective intelligence known in history. And what to me is exciting about it is there's a sense that oh, we know the answers already, and I think that's a very bad view. This sense that like, "Oh..."

James Cham (06:41):

By the way, that is bad for the technical utopians who feel like who are not humble about understanding what the future could look like, but also it's bad for the critics who don't realize that these things all can change, that the fact that it's possible that 500 years from now, we're still stuck on Unix and the internet, right? It's also possible that we won't be, right? We're still so early and with that willingness to be humble about how that change could happen, and that openness to the next great angle, or the next great founder, that's the exciting thing about this stage.

Peter Wang (07:18):

To use a different metaphor, you're Morpheus looking for the Neos that can see through the matrix, right? Because if you are merely a conventional user of these things, you're born into a world, someone drops a smartphone in your hand. You're still stuck sucking your thumb, you're looking at a smartphone. You're like everything that's here is here. Just like you would grow up in a city, and you're surrounded by walls and there's pavement. Underneath the pavement, there's subways, but someone dug the tunnels, someone ran all the steam pipes. Someone built these giant concrete forms on which the elevated roadways are built. Everything you're born into you feel is normal.

Peter Wang (07:51):

I'm in my early 40s. There's a generation of us who grew up in an analog world and watched the digital world emerge. And we could remember when no, you really had to flip through a book to find someone's phone number. You had to dial the phone number, sometimes on a rotary phone. And it was all just wires and electricity impulses. For those of us who are a little bit later, I mean mid-40s I guess is not young, but it's not old either. But in internet years, it's really old because I remember all the old stuff,

right? For us, we can look at it and say, "Yeah, of course it's all just construct, and any of this could change. And it could be Unix, it could be Windows, it could be whatever."

Peter Wang (08:32):

Yeah, all these things are just conventions. Is it a file system? Is it a directory? Is it a database? Who cares? These are all just conventions. But if you're a software developer three, four years into the job, you're trying to learn 50,000 different acronyms and concepts. For you, they're all predefined. They're parts of the labyrinth that have already been laid out when you showed up, and you don't realize that it's all construct It's all just essentially coherent inner subjectivity. It's coherent agreed-upon protocols, and you could make different ones if you want to. And I think somehow we've lost that to your point about it's bad for the techno utopians who start with these imaginary things and believe they're foundational, when any of them could change on a dime.

Peter Wang (09:18):

And it's also bad for the critics because they feel like in a sense, they're reacting to things as if they're solid, when they're not solid. They're all just conventions.

James Cham (09:26):

I think in that way, many of the critics who I respect a lot and who I feel like I respect for both thinking deeply about the questions, and for examining in a critical light a lot of the technical progress we've made, I respect that, but I do worry a lot that they assume that the world is static. That part I worry about. But by the same token, let me just highlight one other thing that you said, which is I think the fun thing about my job is that I need to be humble enough to be open to something different, but I am looking for people with crazy conviction.

Peter Wang (09:58):

Crazy conviction, all right.

James Cham (09:59):

Ironically, right? And then on my side, I'm providing enough faith in them to either work with them, or provide concrete capital. The irony is I need to be humble about my openness to this thing. But to some extent, I'm also looking for founders who are not that humble, who are crazy enough and believe enough to say that, "No, no, no, I either see this future so clearly that it's going to happen and it's going to be inevitable, I want to be part of it, or I see this future that I want so badly that I'm going to devote either the rest of my life, or the next few years on making this happen." And that disconnect is often interesting.

Peter Wang (10:37):

Well, I think humility and I'll push back on just a little bit of the framing you have there, I think humility and vision or conviction are not antonyms. And the best sense of humility in my view is openness. Now, we think sometimes of humility in terms of openness to the idea you might be wrong. But in general, it's an openness, openness that the world might have something new to offer to you. The openness that someone else in this room might have something to offer, even though you think you're an expert on the

topic, right? Even within the confines of what we traditionally think of someone who's very arrogant versus humble.

Peter Wang (11:12):

But outside of that, the generalized concept of humility is an openness to things being different than your epistemic boundary. To some extent, you can build conviction on that as well, that you're open to the world being different than it is, and not only that. But in addition to that, you have conviction that it could be different in this particular way, that would be much, much better, much more interesting. I don't think they're necessarily antonyms in the way that you position them as potentially antonyms, but I love that concept.

Peter Wang (11:42):

To then build on the Morpheus, Neo metaphor and to say let's find a different metaphor than farming, you are Morpheus grooming many different Neos to bash themselves into various things to see if they really are the matrix, or if it actually is a solid wall, something like that. I don't know, maybe the word grooming is too harsh.

James Cham (12:02):

I think one of the great things about the world we live in right now is that VCs get to do a lot of marketing. You hear more about VCs. Then as a founder, you get to get a sense of what they're like, and maybe then have a slightly better chance of choosing to work with someone you like. So, that's good. The bad part is if you take any of it seriously. Because at the core of the entrepreneurial experience is the founder creates the value, the team creates the value. And if you end up treating your VC as a Yoda, then you've failed. Your VC is just like one person in your ecosystem that provides assistance and perspective, and maybe injects either some energy or some capital, or some time along the way, but the core of it is going to be you, and you know this, right?

James Cham (12:52):

Even the best most famous VCs in the world, they're still there providing the perspective, but not actually doing that core work and that core hard work of creating the future.

Peter Wang (13:02):

They're providing the capital at the end of the day. They're very honest, it's venture capital, it's high risk capital management, but there's something else you said too, which is when you're looking at people who can imagine different kinds of software architectures. Now obviously from the perspective as an investor, it's not just pure software architecture nerding out. You want to see new kinds of businesses emerge. Ideally, higher productivity tools lead to different kinds of business models. One thing I do get, I mean you're very right in saying that VCs have a lot of space and time maybe to market themselves. And what I've seen is there's sometimes a narrative of like, "Oh, here's a market opportunity emerging, or we invest in this kind of business model or whatever."

Peter Wang (13:51):

But as a software guy and as a software architect and someone who likes to visioneer things that don't exist, for me, it always seems like software architectures and business models co-create each other. Neither of these exist in a vacuum. Do you have any thoughts on that? How does that hit you?

James Cham (14:08):

Yeah, there's always a co-creation dynamic, but I disagree. If you're going to say what's the chicken and the egg, what's first, I'd say something a little bit different. I'd say that actually not just technical architectures, but the development process for specific technical architectures, not just the development process narrowly speaking around how you write some code and check it in, but how you change your understanding of requirements, how you write the code, check it in, deploy it. That cycle is different depending on technical architectures, and then that cycle then has different characteristics, economic characteristics. And when you have different economic characteristics, then different business models lend themselves to those development characteristics.

James Cham (14:56):

So, that's why if you were to look, and you look at long arc of history, 60 years, it's not that long. And then you'd say the process for IBM building Sabre is very different than the process of the 24-year-old building the latest machine learning architecture. And because that process is different, the way to deliver value is different and thus, they're different business models that become enabled. If you talk to someone who's a developer in the, I don't know, I guess the '80s, the idea that you have constantly upgrading software is crazy, because the first idea would be that the software's always available, right? I'm an investor in this company called LaunchDarkly, right?

James Cham (15:43):

That sort of shift, that only makes sense if you have always-available software and the chance to update at any point, right? And then that only makes sense in the case where you have this entire infrastructure for developing software online. And then that part only makes sense if you've done all this work. Those are the slightly crazy shifts, which you talk to someone who might have retired from software development 30 years ago. They're like, "This is crazy." And my guess is this is the speculative part, which is my guess is we now live in a world where people take data seriously in a way that they really never have before. And that's in the same way that we moved to this always-available software had all these implications, this world in which we take data seriously.

James Cham (16:31):

And I'll tell you, I'm going to be snarky about it, but I'll describe what I mean in a second. But in that world that there are new business models that make sense for that world. Okay, so what do I mean when I say we take data seriously? What I mean by that is if I built at some point some warehouse management system, it was great. I was super proud of myself, but the truth is that we really used that data for two reasons. One reason we used the data was to make sure that the government understood how much money they were making, so they can then write off whatever for taxes. We really did for regulatory reasons to show the government that we're paying the right amount of taxes. I mean ultimately, go down the number of steps.

James Cham (17:11):

That or to create data points that would support the argument of some middle manager who wanted to do something. Meaning, that it was really either the data was, we were said we used the data for data-driven decision making, but that wasn't really true, right? And the profound shift that happened over the last decade or so is that there are a whole set of companies that only could work at scale if you took data incredibly seriously. Because otherwise, you couldn't, I don't know, whatever. You can do the matchmaking to find the car, to make the offer, to have someone drive or whatever. All those sorts of matchmaking things, all the commerce, all those pricing decisions, all those sorts of things, actually they were a fundamental shift.

James Cham (17:55):

And in that case, that's why you have as you know, the rise of a whole set of companies that worry about data quality, or think about data definitions in a very different way, in a way in which it feels much more life and death. And thus, life and death for the company, and sometimes life and death for healthcare. In that world, we're just figuring out that that's how we change our point of view. And then what we haven't done yet is figured out all the business models that lend themselves to that change.

Peter Wang (18:26):

There's a lot to unpack there, but I fundamentally agree with you that for all the talk about big data, for how much of the world runs on data systems, for the most part, data is still a static almost post... It's a thing that's produced by processes, as opposed to being the substrate in which informationing happens. And one of my absolute favorite pieces that I've ever read about information and this kind of stuff is the Selling Wine Without Bottles. It's on the EFF website. We'll add it to the show notes here. I recommend everyone read it, but yeah, Selling Wine Without Bottles by John Perry Barlow. And it's about the economy of mind on the global net.

Peter Wang (19:05):

And that's where he articulates some of Greg Bateson's ideas about information is a verb. It's the dance, not the dancer. The thing about what you're talking about with we haven't yet figured out what software's really supposed to do, and what it's good for. But now as we think about moving to a world where in a data-rich world, data is the substrate of which would build new business models. And that's different data, it's not transactional data. I'm sorry, it's not data for the purposes of just transacting into a database. It's that motion software doing that and the data systems that support that, that's still a, what is it? It's like a record keeping system. It's long-range telepathy, right?

Peter Wang (19:45):

Deeper telepathy between people, the systems, and whatnot, but what you're talking about is really decisioning, informationing, predicting all these kinds of things that are only possible once you have a data-rich environment, once you're able to observe all these things, to surveil all these things about the world, or sometimes creepily about people. Once you have those things online, then an entirely different set of business models appear. So, that's my take on that.

James Cham (20:11):

Right. And then the other shift though is that there's a way in which you and I because we've seen the before times are just amazed that this even works. We're just shocked. I mean the part of me, it's all just really impressed that all these things get strung together. And it's going to be the set of folks who see

that and take it for granted and say, "Ugh, I don't care that you managed to string this together, or that your database was consistent. Who cares? I'm unhappy about the fact that I can't do this and this and this." And that's what you're looking for as well, right? You're looking for the set of folks who then could say, "Oh, this fits differently, and we need to twist it in this way because I'm dissatisfied, or I think it's ridiculous that you have these such low expectations." I think that that's...

Peter Wang (20:58):

Yeah, you're looking for the people who can see that the walls are made of just light, and you're also looking for creatures who can just become creatures of light, beings of pure energy that bounce around, making TikToks of themselves as these ethereal creatures and getting millions of influencers. The other thing though tied to what we said earlier around... you talked about it's not just the software architectures. It's the process of doing product development. It's the process of how do I engage with my users, about the customers, or potential customers, find out their needs, and then build a thing and that interaction. In a sense, it's what a generalized Conway's law, because Conway's law was about the internal development structures.

Peter Wang (21:39):

How you lay out different development teams and their communication patterns defines the software architecture. It defines the internals of a thing, but what you're talking about a generalization of this, which is how businesses, how product development teams engage with their users. That process and the cadence there and the possibility and exposure surface there, that defines to some extent the different kinds of business models that are available.

James Cham (22:02):

It's not that it's a generalization. It's probably like another level up of abstraction. Ultimately, industry structure is going to be driven by these changes in technical architecture, and these changes in technical architecture follow some grain-

Peter Wang (22:18):

Got it, got it, got it. Yeah.

James Cham (22:18):

... of the way that development works today, right? That's probably the way that I think about it. And then the other thing about it that's funny is that, or my big disappointment is that the ideas should be obvious and then the implementations should be hard, but there's still not enough variation in the ideas. And I think that's partly because we're maybe as a culture, a little too worried about harm now and thus, we're not willing to imagine a little bit differently. So, that's one part. And then I think the other part is that propagation of folks who understand data or understand machine learning, they haven't propagated up to more leadership roles, right?

James Cham (23:00):

You're seeing that, but you're not seeing enough of that. And then the other part is this assumption that the technology is... I'm trying to think of a good example. If you read science fiction stories in the '50s and they talk about mainframes and like, "Oh, the mainframe will save everything, or the mainframe will

control the world, or the mainframe will do this, this, that" and you read it now, it's obviously ridiculous. And we're going to think the same thing about an awful lot of writing about machine learning.

Peter Wang (23:25):

Yeah. I was listening to it a podcast last week, and this guy's making a great point that we think of informationing, we still have a very computational paradigm for decisioning, informationing, et cetera. And he was making the argument, he's trying to make the case that thinking is actually an outgrowth of feeling. And that when you have conflicting feelings, then you actually have to think to resolve the conflict between the feelings. Otherwise, you can want an autopilot just on feelings and here, we are building thinking machines before we build feeling machines. And maybe is that actually always doomed to fail as an architecture, or rather it can only ever be a shallow imprint of whatever paradigm the human architect put into the system, right?

Peter Wang (24:04):

That for it to actually develop organic something aspirational beyond the boundaries of the sandbox we build it on, for it to build beyond it, the system internally has to have conflict that forces then free energy to build the next level of structure, which beyond emotion is thought, right? Something like that.

James Cham (24:23):

I don't know whether the right distinction between emotion and thinking. It might be system one and system two. It might be that snap decision, versus the reason decision. And certainly, you see some of that in companies, where you think about it in terms of escalation, right? You have processes and a bunch of business rules coded. And sometimes, you need to bubble it up because there's enough problems. And then okay, so now this is the delicious part. I think the delicious part to me is that right now, by calling something an algorithm, we concede too much. And rather, the right attitude should be exactly what you said, which is there'll be times when the snap judgment, or the business rule, or the machine learning model will be wrong.

James Cham (25:09):

And understanding all the trick is not around getting the machine learning model which be right more often. The entire trick is around finding problems in which when the machine learning model is right, you benefit a lot. And when it's wrong, you don't get hurt that much, right? So, that's a business model problem definition thing, or creating a system in which you understand that it'll be wrong sometimes, and you understand how to deal with it. In that case, that's a bureaucratic question, not a philosophical question, not a technical question. We're dealing with some of these questions at the wrong level. And at this point, I will have to pitch my two favorite AI pieces.

James Cham (25:47):

There's a guy named Henry Farrell, who's mostly a foreign affairs guy, who at some point wrote this piece of Seeing Like a Finite State Algorithm, no, Seeing Like an Algorithm in which he basically does a very good job of thinking about the fact that these models are based on data. The data is often generated by people. People have incentives, and they will lie and cheat and try to fool the model if it's in their self-interest. And it's not just around scary things, hate speech or stuff like that, but it's also about simple things like making sure that your electronic medical record gets approved, so you get billing approved for your doctor's appointment, right?

James Cham (26:28):

So, that dynamic nature, that's one interesting one. And in fact, it's baked in. You're just going to have to assume that people are going to be clever, and we're going to come at ways to cheat the system. So, that's one. And then the other one is Street-Level Algorithms which is written by these two Stanford guys. And in that case, it makes the point that the right way to think about machine learning systems, the right way to think about is probably more to think about bureaucracies. And there's lots to be learned from how bureaucracies are managed.

James Cham (26:57):

And rather than, thinking in terms of, "Oh maybe, we'll be finally be able to construct our god-king machine learning model to do AGI," the right way to think about it is, "Maybe we'll finally be able to do another example of an interesting organizational shortcut and trick in order to make things a little more efficient." That's the right, but the model will be wrong. The rules are wrong sometimes, and do we have systems to blanket that and to deal with it? To me, that's the way to think about it.

Peter Wang (27:23):

So as a synthesis of this stuff. Tell me if I'm correct on this, what I'm hearing is that we should be using machine learning and data systems, everything. We should be humble about it, and apply it to system one things to free us up to actually address system two for now at a human level. And then maybe if we develop competence there at running system one things at scale and with high quality, high fidelity, then we can maybe allow them to come into system two. But to try to take system one, system two together, put the whole thing and just say, "Well, here's a pile of data, here's a pile of GPUs, make the whole decision, and let's evaluate the whole thing," that's doomed to fail. Is that a reasonable synthesis of what you're saying?

James Cham (28:05):

Another angle might be something around there's this assumption that if only we get the models right, perfect, then we'll be fine. And the assumption's you got to assume nothing's perfect. And then the funny thing is the step before that is, "Oh, only we get our data perfect." I'm like, "The whole point of machine learning is the data's not perfect." It's interesting to me that and then what's a meta theme there, there's something about diminishing returns around trying to go for the 99 point whatever. How many 9s do you want, or do you want a system in which you assume that you're not going to get eight 9s or seven 9s? Does that make sense?

Peter Wang (28:41):

It does, it does. And as I hear you talk about this, I can hear C-suite folks who are decision makers, budget holders and budget approvers who have this mentality that it's almost like they look at computer-augmented thinking and decisioning. They want to think of it as an industrial process. And to your point about calling back to the '50s or even to steampunk era, when people are thinking about industrial processes, you build big machines. You put a ton of steam power into it. You build these giant things that all moves, and it moves fast and it's hugely powerful, but there's almost a simple naïve, homogeneity, cleanliness to that thing. In the world of ML and AI or data science approaches, these are tools to be used in spaces of uncertainty. These are used when you go outside the bubble.

Peter Wang (29:36):

These are tools to go into the outdoors, not to be used inside the nice, clean city walls where the data's clean, the water's clean, everything's sanitary. This idea that when you have the buyers as a flip or a dual of this idea of Conway's law and building different business models, how businesses want to procure, how businesses want to think about what is actually knowledge work, that informs also what possibilities of things you could sell to them. And everyone thinks about tools. They want tools to make better data, tools to label their data, tools to tell them when their machine learning model has gone awry.

Peter Wang (30:12):

Everyone is focused on the tools and this industrial way of approaching this, almost as industrial process. When the question I think should be carving out apriori, what are the human elements? Where are the spaces for humans to wield the tools? And from that position on...

James Cham (30:29):

On that point, I might disagree. I actually think that the thing about industrial processes is that they assumed that things were not perfect. And then the thing is that the best executives and the most successful ones had a clear understanding of their own industrial processes. They spent a bunch of time in understanding it and decomposing it, and getting it to work. They had mastery as much as you could have mastery over it, as opposed to conceding it to some small part of the organization, or to another company, or even worse to consultants. And in some ways, I feel like I'm actually suggesting the opposite, right? I'm suggesting that no, you as executive, you need to really understand it.

James Cham (31:12):

And you can't treat it the legal team, or something like that. In some ways, the best executives are going to be the ones who really internalize it and understand what's possible and not possible, and have the right mix of skepticism and willingness to understand when it does something really, really well, understanding the grain of the product. And I think that's one part. And then the other part as you talk about it, I realize one of the things that you and I joke about is at one point, I used to write that I'm looking for tools for a more elegant age, right? Like the lightsaber, right? And I wonder whether I'm wrong about that. I wonder whether the look for a tool is super individualistic.

James Cham (31:54):

It's like, "Oh, I could be the Jedi with the lightsaber to save the organization." And I wonder whether the real search should be for organizational-scale services and products that could pull things off. There's a little bit of a what's the real problem in Star Wars? Is the real problem that the real problem could have been the Jedis and the fact that they wanted to have these lightsabers, and be able to save the world individually, rather than dealing with things at a systemic level.

Peter Wang (32:23):

Well, it's the Maslow's hammer problem, right? If you have a lightsaber, you want everything in the world to look like a bespoke problem, that needs a Jedi to drop in with a lightsaber.

James Cham (32:33):

Maybe the answer is we need bespoke organizations. I've poked around this idea, and maybe you can help me articulate this better. I poked around this idea that I want to invest in organizational-scale user

experiences in companies that think of themselves not as tools for thinking. But in services, they think about the user experience not just from a one person-to-person point of view, but somehow have a bigger view of things. And I don't know exactly what that means. I get tastes of that. I've invested, or we've invested in a number of companies that are all around employee empowerment and stuff like that, right? But the real goal there is to somehow get an intuition around what software end up of creating these think about changing things on an organizational level.

James Cham (33:21):

And that part, I feel like I'm still grappling around for that.

Peter Wang (33:25):

Well, it's exactly the right question, and the word I would use is collective sense making that you're absolutely right. This is the kind of conversation we would have late at night as it goes off into the early morning.

James Cham (33:38):

I feel a little guilty. I feel like what I should be saying is whatever, data stack and I should be...

Peter Wang (33:44):

Right, you're going to invest in the modern data stack.

James Cham (33:46):

Low code, no code or machine learning, NLU or no, I feel like I should be stuffing this conversation in with the various buzzwords. So when the transcript is produced, it'll show up better SEO for venture capital purposes, or for finding founders, or for you entertaining your audience, but I do think these are the deeper interesting questions.

Peter Wang (34:05):

Well, what we want to do is in this podcast, hopefully every single conversation, it goes just one tick more philosophical than people are maybe comfortable, or interested in. It goes one tick more technical. It goes one tick more into business versus the not. We're going to do all of it, and that's just we're going to do this thing, or see if there are people who inhabit the longer tail couple of standard deviations out from the mean. But to that point, but we really get to the heart of a very interesting thing, which is what I'm hearing is you want businesses to think about decisioning systems, real information systems as different than mere record keeping systems, like a Sabre or something.

Peter Wang (34:45):

And if there's anyone from Sabre listening, I don't mean to denigrate what you're doing, but it's just more of like there are a lot of transactional systems that are the successes of so-called information technology over the last 50 or 60 years, but a lot of it is database crud. It's a lot of it aggregations, summing things, basically summing things and counting things, and doing very simple linear regression, but what we're coming into is this problem of okay, if you think about the big businesses all over the world, a lot of them treat technology as a cost center. Information technology is of course you got to

have. You got to have email. You got to have people doing spreadsheets to figure out how much money you made, or how much money you lost.

Peter Wang (35:24):

But at the end of the day, they're not even system one thinking a lot of times. They are really just almost like sensor systems. They're just telling you which pixels are green or red in your financial landscape, but decisioning still all happens at a very human level. It happens over a lot of emails and calendar invites and talk and people talk. And sometimes, people write things down, but all the actual collective sense making, we use these information systems as basic just sensor systems. Now, when we flip to decisioning systems, automated decisioning systems, as they have at layer one.

Peter Wang (35:57):

And then the deeper question you ask, which is these executives actually becoming aware of the fact that not only are they aware that the grain is important, the grain of the wood that's going through their mills, but actually to understand your job is to actually be aware of that, because that's the quality of what goes into your entire... Every business is an information business.

James Cham (36:18):

Okay. Let me give you my concrete examples of the inversions that I'm interested in that I just haven't seen. You've dealt with various annotation systems, right? Either to say that something's a bridge and an image or to highlight some text to say that it's a dog, or it's an entity, or it's whatever, financial transaction. What's interesting to me is that sort of structuring of work is only done right now for people who get paid less than $10 an hour. That vast majority of all that work of all the tools for that work, and all that work really is being done and viewed as simple, snap judgment-y sort of things.

James Cham (36:58):

I think that there's a great deal of value in structuring the work and making legible the work that knowledge workers and these slightly loosey-goosey, unstructured people, like you and me, in the sense that we're executives. We think we have lots of tasks. We think we're really special, but I bet you there are a whole set of things that we do that really could be boiled down and digitized, and structured in the same way that you'd structure something for Mechanical Turk. And that thing, that seems obvious to me. Now, the part that's not obvious to me is how to do it in a way, that big ego executive doesn't think of themselves as low value. How do you describe it in a way that is status enhancing?

James Cham (37:46):

How do you quickly create lots and lots of processes that capture the fact that there are all these corner cases? That whole mix, that's the hard part, right? And in some ways, the way that I characterize it is I am looking for tools, for annotators who don't get paid $10 an hour, but get paid a thousand dollars an hour, right? That's an example then of something I think is just right out there. I think that I can smell the opportunity. I can't build it, I can't describe it. And thus, I'm sitting around hoping to the universe that someone will do it, but also I'm waiting for the moment that I meet some founder. And they say, "I got this thing." And I'm like, "Oh, this is what you're doing."

James Cham (38:27):

And they say, "Oh, you know what, actually that is right, that is what I'm doing." And then we get along and then they say, 'We will take your money even though your money is so expensive." And then we work together and then five years from now, we're super famous and have some conference with Peter.

Peter Wang (38:43):

Let's turn the tables. We'll see how the tables turn here. I'll play the VC to your pitch on that thing, and I would say this: If you actually had something like a whiff of a technology that could automate a way, the hundred, not even thousand, but $100 to $500 an hour kinds of executive-level, super white-collar tasks, the way you actually make money on this is you don't go to the executives. If you sell them some automated decisioning tool or something like this and you get it wrong, you're fired instantly, because the downside is so hard. You and I know this, all the executives, all the whatever people at the top of their game and all this stuff, they're surrounded by, supported by a next tier of manda... I call it mandarins.

Peter Wang (39:25):

Maybe that's not a common term, but there's a tier of ministers. And the ministers are super worried about losing their jobs. What you do is you produce the prospect of an automated tool that takes their jobs away, but what you really do is you sell them the protection to keep their jobs from being taken away. It's a very cynical flip on this.

James Cham (39:47):

Now, I actually have a number of variations. I think the way that that core insight gets implemented in the business model, I think the actual product, there are lots of variations. But just to be clear, I'm actually saying something a little bit different. I'm saying that that way of structuring and describing and thinking about the work is really valuable. What the annotator ends up doing to try to flatten something into some vector, ultimately to flatten something to some vector, that work is actually interesting and valuable. It doesn't necessarily have to lend itself to a machine learning model that automates everything away, but just that way of thinking about the work that you're doing could be interesting.

James Cham (40:30):

There's a company out of Boston called Resemble that was doing something like this, Reassemble, that was doing something this for salespeople where it'd say, "You write up your sales report. You highlight the 15 things in your unstructured text that here are the two objections, here are the three opportunities. And then we will know that that's where it is in the text, and then we will unfold it in some big table. Then you can see for every customer, here are their objections." That sort of work isn't necessarily automating something away, but it's taking advantage of all the learnings we've got around tool making for annotators and applying that for people who aren't necessarily just getting paid $10 an hour or $2 an hour. I think that's one of the angles, but yes, there are many other angles on it, and I'm super curious.

Peter Wang (41:15):

Do you think Google will get there first with email auto complete basically at the end of the day, is Gmail auto complete going to eat this entire category because they literally... I mean they've got everybody's email. Holy crap, no one's talking about this, or people are not talking loudly enough about this, but it's

like that's an amazing corpus. That is an absolutely amazing corpus. Now, I know they have safeguards, but...

James Cham (41:37):

It is really impressive.

Peter Wang (41:39):

I mean Salesforce similarly, right? All these 2005 to 2010s era SaaSs, they have a lot of customer data.

James Cham (41:47):

Well, okay, okay. They do and they don't. I will remind you of the electronic medical record problem, which is I as the hospital or organization or the insurance company, I might want to learn a bunch of stuff at scale about what treatments work and don't work. But the truth is the ground level annotator or the data provider which is say this doctor, the doctor just wants to bill approved, right? I mean they're going to be honest, but they're going to write the description of it for a purpose that's at cross purposes from what the organization might think about. And I think being clear headed about that, that's really valuable. And that's the sort of thing that I think the smartest people are just starting to get. And I think that that's the opportunity.

Peter Wang (42:35):

Yeah, we have now roamed far and wide over all of this. But at the end of the day, we come back down to you've got a fund. You're looking to invest in people who have the insight to see that the world isn't just the way it's constructed today, that it could be something radically different, but that they also want to transform and change how people are able to work together, how people are able to net produce more collective intelligence beyond the traditional firm structures. That's what I'm hearing this underlying sentiment, a lot of things you're talking about.

James Cham (43:07):

Or another way to put it is if you are ridiculous and ambitious and crazy enough to listen to Peter all the way through these podcasts, then I'm interested in chatting with you.

Peter Wang (43:20):

I'm humbled and flattered James you do a great honor, coming on here and having this great conversation. I'm sure we'll have many more. We're out of time today, but the thought that I wanted to... we didn't talk about PyScript.

James Cham (43:30):

We didn't talk about PyScript which is one of the [inaudible 00:43:32]...

Peter Wang (43:31):

Maybe we'll talk about PyScript in the follow-on, right? But the thing I would put in your brain to think about is a lot of the stuff you talk about touches on in my mind one of the deepest things in statistics, and one of the deep truths we have just of in metaphysics is the bias variance trade off, that we can

either manage variance away, so we can better produce here's a clear offset, here's where things are at, or we can be tolerant of systems where there's a lot more variance, but we've got to be okay with there being a lot more variance. If we can get anything on the universe, we can get a little bit more upside than downside, then we're making progress, right?

James Cham (44:08):

We also didn't talk about my two other bugaboos or my two other things. One is we didn't talk about explore exploit trade-offs in systems and in people and in decision making. And we also didn't talk about my current silver bullet, which is clearer thinking about error correction. I'm convinced that error correction is not just a key to evolution, but it's a key to all human progress.

Peter Wang (44:33):

Thank you. Next time, we'll talk about that. That takes us much faster into a metaphysical discussion, but a very fun one. Thank you James. Thank you so much, really appreciate the conversation. And in the show notes, we'll have links to some of the things that we brought up and the articles that you mentioned, in particular that sound very interesting. Thanks to everyone for listening, and keep an eye out for the next episode, and keep an eye out for the part two of the Peter and James nerding out on the metaphysics of learning and decisioning.

Peter Wang (45:03):

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 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 anconda.com for more information.