How to Build a Multi-Billion-Dollar Software Business | Mohit Aron, Founder of Cohesity

68 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=HP2OGGNT7DC
https://www.youtube.com/watch?v=hP2OggnT7dc

SUMMARY

Mohit Aron, founder of Cohesity and co-founder of Nutanix, shares insights on the startup journey, emphasizing the importance of a well-structured hypothesis document for validating business ideas. He discusses the critical stages of building a startup, from achieving product-market fit to scaling operations, and highlights the transformative potential of generative AI in various industries.

- *A well-crafted hypothesis document should include an elevator pitch, minimum viable product (MVP), competitive advantages, and potential objections (why nots) with rebuttals.*
- *Achieving product-market fit is defined as when average salespeople can sell to average customers without involving founders, indicating repeatable sales processes.*
- *Founders should be open to feedback and surround themselves with high-quality advisors to avoid confirmation bias and ensure honest assessments of their ideas.*
- *Effective hiring and performance management are crucial; using structured scorecards and OKRs can help track progress and identify underperformers.*
- *Generative AI presents significant opportunities for disruption across various sectors, encouraging founders to integrate it into their business models.*
- *Continuous learning about sales and go-to-market strategies is essential for technical founders transitioning to leadership roles.*
- *The future of technology is unpredictable, but advancements in AI and biotechnology hold great promise for transformative innovations.*

I I I think it's a blessing for people not to get funded when they're doing this kind of stuff, rather than they do get funded. They actually do intend tend to convince someone somewhere to give them the money. And then they go through four six years of hell and then they see, "Oh my god, I didn't realize, you know, my six years of my life has gone to waste. Uh why am I so unlucky? You know, why are so many other companies successful? Why is it It's like then they've gone into uh you know, self-flagging and um self-pitying and whatnot. It's just Surround yourself with great people, high-quality investors, high-quality advisors, and they will tell you the reality, right? If you're off, they will tell you. But if you keep, you know, drinking your own Kool-Aid and you're like, "These guys are wrong. I will show them."

Okay, yeah, sometimes it happens that people are wrong. But how likely is it that good people they're all wrong and you're the only one right? From Foundation Capital, this is B2B CEO, a podcast about the startup journey about going from idea to IPO and growing from a founder into a CEO.

On each episode, I speak with notable CEOs and founders and get their stories about what it took to build a company of scale and become a leader in the enterprise. I'm Ashu Garg, a general partner at Foundation Capital. Today, I'm thrilled to welcome Mohit Aron back to the B2B CEO podcast for the second time. As the founder of Cohesity and the co-founder of Nutanix, Mohit is a titan in the worlds of both enterprise go-to-market and infrastructure software.

With two widely successful companies under his belt, he's a true expert when it comes to building enterprise software businesses from the ground up. In our conversation, Mohit shares his proven frameworks for validating startup ideas. He reveals hard-learned lessons from starting Nutanix and Cohesity with real-world examples that bring his advice to life. We explore product market fit, what it really looks like in practice, as well as how to build a team and manage performance in a high-growth startup.

We wrap by discussing the topic de jour, generative AI and the opportunities it opens for startups. This episode is packed with insights for technical founders. I hope you enjoy it. Mohit, thank you so much for joining us on this episode of BWB CEO. Happy to be here, Ashu. Thanks for having me again. Having co-founded both Nutanix and Cohesity and being an early employee at several other startups, uh I wonder, have you Do you have a framework for how you think about starting a new company?

What I recommend any founder to do is to write a hypothesis document on why their company would succeed. And here are the key sections, if you may, in that hypothesis document. The first section is uh basically your elevator pitch. If you were to run into either the CIO or whoever is the buyer at a potential customer, what would you say to that person to um kind of motivate them to look more into what you're selling? Right? Um so, that's the first section.

The second section is your MVP, minimum viable product. Now, I've heard founders they talk about a big game on what they might do 5 years from today, but their their company is going to take off on the first product that you sell. What exactly is that? Crisply define that. So, that's the second section. The third section is why would this product Why would your company be successful against any other competitors that you may have?

So, write down why you would be successful in your mind. And the last section is equally important, uh which is why not? So, if No, look, all of us founders, we like to drink our Kool-Aid. Put yourself in the shoes of an naysayer and write down what a naysayer would say on why the product will not succeed, why the company will not succeed. And you need to write them down, the why nots, and then you need to have a convincing rebuttal against all those points.

If you build this hypothesis document and you're honest, intellectually honest, I can nearly guarantee that your company will be a success. I do have a checklist of things that I consult that inform the whys and the why nots. For instance, is the company operating in a big TAM? You probably don't want to do a company in a small TAM. Is the company, you know, addressing an immediate key pain point? I mean, is it a aspirin or a vitamin? Vitamins are good to have.

Aspirin is a must-have. So, some of these things are checklist items that you can consult while writing the whys and the why nots, but the framework applies to every company. Build this hypothesis document. I'm happy to kind of run the the document, you know, through the two companies I founded. That would be great. Maybe just start with, you know, which one do you want to start with? Yeah, let's do the first one, Nutanix. Right? So, so the elevator pitch, um you know, the traditional legacy, you had to go to three different vendors, compute, storage, and network vendors to put together a a platform on which you can run your applications.

And Nutanix came by, you know, with a platform that collapses all of these. Uh so, it's hyper-converged. Uh and so, you don't have to go to three different vendors. It saves you cost. Much easier to manage. That's the elevator pitch. Right? Um the second section, as I said, was the MVP. What's the minimum viable product? The minimum viable product was something that ran as a distributed system that basically lent itself to virtualization. You can run VMs on it.

And it addressed VDI. You can run virtual desktops. That's it. We didn't want to do anything more. That was the MVP. It was a one-stop shop. You didn't have to go buy anything else from anyone else. You just come to Nutanix, you get the whole shebang. Then the whys. Some of the whys came from the hardware trends. Um you know, the CPU speeds and and the disk speeds had been increasing year over year, but the network speeds have pretty much been not increasing, you know, in the last two decades. So, it made all the sense to kind of collapse and bring compute and storage together and take networking out of the way. That was a big why.

Uh another big why was it was a big market, right? The TAM, you know, just for VDI was at least a couple of billion dollars. Right? I think somewhere uh north of two to three billion dollars. Right? Um and the pain was immediate. Um anyone who uh people who wanted to build virtual desktops, they were um basically saying that building a VDI environment was so expensive that they'd much rather buy laptops for their people. Right? So, they were the pain was immediate.

So, we ran uh a bunch of these whys and and that was what was going for us. And the why nots, I mean, the naysayers were saying that well, uh SANs are already there. They have a big penetration in the market. Uh why would you succeed? And the answer was, well, if the pain is high enough, uh people are already looking for a solution, they would buy. So, there were a bunch of these why nots and we would talk to customers and come up with the potential why nots or come, you know, whoever else had a why not, uh they were a naysayer, we would write that down and we had a counter to that. Uh on why we think, uh you know, objectively that that why not does not hold. We had a rebuttal for that.

And it was a pretty, you know, pretty tight document, if you think about it. Um you know, basically told you that Nutanix is going to succeed. That And that's basically what happened. And you know, and you fast forward today, I mean Nutanix is a $15 billion public company. You know, maybe you could walk us through the journey of how you went from being successful in what was already a large market, VDI, uh to much broader platform. Was it always at the back of your mind?

No, so that's actually, uh you know, one of the things in my playbook, which is your MVP should not be equal to

your vision. Because if it is, you may be the first mover on the MVP, and you will have success for a short amount of time. But other people are also watching, and they will copy. And by the way, whoever is listening, please understand that patents don't protect much. It's just a defensive strategy. Try going into a patent war with a big company like, you know, EMC or what have you. It's you're just going to settle.

Um you're not going to win. So, uh you need to My my take is that you need to have a much bigger vision that you can then iterate on. So, you start with that MVP, and then you iterate towards a bigger vision. Uh and then people, even if they try to copy you, you're further ahead by iterating on that vision. So, one of the reasons, uh one of the whys also is, you know, there is a bigger vision.

Right? So, if any naysayer says, "Well, what if uh competitor uh copies you in a year?" So, they give you a why not. Your rebuttal is that well, I have a bigger vision that I will iterate on and I'll get ahead of the game by the time they copy me. So, Nutanix's biggest bigger story was of course the story of hyperconvergence where it's not just about VDI, but they will also hyperconverge other stuff like databases and a whole lot of other stuff on on this platform.

Got it. And so, they kept we had, you know, it became a big hit at the at the outset and they were at the time some 10 different companies and one of our VCs said that every other day they have a new entrepreneur sitting in their office telling them how they're going to kill Nutanix. And eventually all of them got killed, you know, except for a few notable you know, competitors, everyone else got killed because Nutanix was just so far ahead. It's It's been a phenomenal story, you know, always a rollercoaster, but but a phenomenal story in terms of the scale of impact it's had. I'd love for you to perhaps apply the same framework to Cohesity. Yeah, absolutely.

Elevator pitch, you know, for the MVP is your backup SLAs fail all the time. Backups today don't scale. They're a pain to manage. We'll give you a simple distributed platform. It pay as you grow and it cuts down your cost tremendously and your SLAs are going to be 99% plus. And by the way, we don't stop there. This is a platform for doing more than just backups. You can actually consolidate the rest of your secondary data on the same platform thereby eliminating lots of silos. That was the elevator pitch.

The MVP, the first one, we're just going to do backups for VMware. That's it. Notice that backups are also done for a whole variety of other things like databases and filers and but our MVP was just about VMware. Right? We are going to make VMware backups really really simple for the big enterprise. We were not going towards the lower end of the market which was controlled by Veeam. So, we were very clear what the MVP is attacking.

Right? The third one was why. The why came from the fact that well, you know, I told you some of the pains. You know, the backup SLAs were getting missed. Backups were very expensive. In fact, I actually got motivated into this area by something I heard from a customer at Nutanix. The person said that they they they didn't have backups. They're like backups are more expensive than a Nutanix. Why would I They were asking us, can you somehow eliminate backups?

So, I I knew there's a problem in in backups, right? People find it a pain. Right? It was very pain It was painful to

configure. It was painful to manage. It was very costly. It did not scale out. So, right there was a lot of whys. And then the why nots, of course, were that hey, this is a 30-year-old industry. Right? How would you penetrate this? Backups are very sticky. What if people don't want to move? And again, if you know, we alleviated that. We addressed we we rebutted that by talking to customers.

And customers said, hey, if you give me something better, man, I'm I'm tired of my backup. I want to change. Right? So, we had nice rebuttals against that. So, and then of course, I ran through my checklist. Right? It was a big market. Backup is a huge tens of billions. There was an immediate pain. Right? It It was not a bad neighborhood. Anyone who had a reasonable working product in the backup space was a multi-billion dollar company.

Right? I always look for a one of my checklist items is that the neighborhood ought to be good. If there's a lot of dead bodies, companies starting and then dying in that area, that's not a good sign. Right? So, so a lot of the whys and the why nots kind of worked worked in our favor. And uh Finally, when I looked at the final hypothesis document, it gave me the confidence that this is strong enough for me to do a company here.

By the way, since you mentioned Cohesity, I you know, I should also say that I'm talking about this hypothesis document in the context of forming companies. But as we started building Cohesity, and we started innovating, initially, my team, you know, they adopted Amazon's PRFAQ thing, where, you know, literally Amazon advisor's writing a a press release, and then a FAQ. And then I showed the hypothesis thing to my team, and they liked it so much, and it was so much more practical that they actually discarded the PRFAQ, and even for new features within Cohesity, we are actually using the hypothesis document. Even if it's a small feature a new feature that Cohesity's innovating on, they write down a hypothesis now.

So, it's it's become an essential part of our PRD. And when you look back at that initial hypothesis document, and some of the core assumptions you made, you know, perhaps you can walk us through the each stage of the journey. Maybe the zero to one, one to 10, 10 to 100, but at each step, what was that journey like? So, let's start with zero to one for Cohesity. What did you find you had gotten right? What were some of the lessons you learned along the way?

Yeah, so, uh again, it's all about the MVP. That's zero to one. And our MVP, like I said, was uh backing up VMware, right? Virtualized environments. And uh what we found was that customers were very interested. Um now, we missed adding a few features to the MVP. Uh and that didn't mean that we weren't able to sell, but definitely we weren't able to sell enough.

Like I still remember one of our very big customers, um they um said, "You We love your product, but we're not going to buy it today, because you don't have replication across sites. So, you cannot replicate from one site to the next one. So, I'm not My sites are not protected." So, so that told us that we kind of missed that uh in our MVP. Um so, we built it. Uh and as soon as we built it, not only did that big customer, it was a Fortune 500, um not only did they buy, coincidentally, the the guy there who evaluated us was so impressed, he actually joined us.

So, that was sort of our, you know, almost zero to not one, I our one was pretty fast, but zero to 10 story. Yep. And then beyond 10, it was uh going beyond virtualized backups, going into areas like uh backing up databases, uh going into uh backing up filers, uh backing up, you know, uh native Windows or Linux or what have you, right? But the But the MVP is very important because that's what starts the flywheel.

Yeah. Once the flywheel is start, your sales guys are selling. So, the the nice equation you want to have is that engineering is putting more on the truck, and sales is selling whatever is in the truck. Right? And over time, engineering keeps putting more on the truck, and sales keeps selling what's on the truck. And now, if a competitor even tries to catch up to what's on the truck, well, by the time they catch up, you have more on the truck, and they eventually they just give up.

Right? So, so that's kind of what happened for the bulk of uh the companies in this space that tried to copy us. And then we Of course, uh the vision was to go beyond backups. So, we also started selling filers. So, the same platform can not just be used for backups, but really to consolidate um sort of your non-production data on the same platform, so you don't have to run that on different silos. So, we also sold this to customers who wanted to use it as a filer.

We sold it to customers who wanted to put their test and development environments on this. So, now their secondary data all kept coming on the same environment, and that was in line with our vision to um be able to deduplicate the data, to reduce the data footprint and improve the manageability of all these tons of data and bring it all on one platform. Then we added the cloud, um right? So, uh when we started bulk of our customers, we aimed for the big enterprise.

And the big enterprise was predominantly sitting on prem. And we saw that uh they're really interested in the cloud. Not nobody's talking about or most people are not talking about moving completely to the cloud. But they do want to have some footprint in the cloud. Yep. And so they really needed a hybrid platform that can do both on prem and the cloud. And that's where many of our competitors kind of missed the boat. They would either build something that's completely in the cloud or completely on prem.

And and and we again, uh you know, uh in the spirit of the whys and the why nots, we uh saw that um our customers, big customers, would like to have a hybrid solution. Um and if only these competitors had built a why not, one of the items in the why not would be how would you even sell to big customers? Because big customers have both on prem and cloud. So, wouldn't they want a hybrid solution?

Yep. That would have been a flag for them to not just build a solution for only the cloud or only on prem. So, we were uh very clear that we will build a hybrid solution. The same product works on prem as well as in the cloud and there's a single UI that they can use to manage it all. Uh they don't have to So, it almost drove out of uh the hypothesis document. You know, we kept extending sort of the hypothesis document and it kind of kind of kept falling out of that. I still remember the first time we launched our SaaS service, right? Um that's where I opened up the concept of the hypothesis document to my team.

And the team built this hypothesis document on why our SaaS service would succeed. They built the MVP. They actually built, you know, couple of MVPs, one for the big enterprise, one for the mid market. And then they looked very different. Right? Um what the big enterprise needs can be different from what the mid market needs. And then there was the whys and the why nots. And and lo and behold, as soon as we released our SaaS product, it was a big hit. Because we had kind of thought about all the problems, everything they want, and everything that would stand in the way, right? And we had already addressed it.

So, it was a hit right from day one. Um so, that was it was just amazing to see, you know, all this hypothesis actually in action. Now, can it go wrong? Yeah, it can go wrong if you, you know, if you're not complete in your whys, if you missed out on the whys, if you did something wrong in your why nots, if if you you can always do that, but the probability, if you methodically think about the whys and the why nots, the probability of that happening reduces tremendously. And that's what it's all about, right? But I'm going to probe a little bit on, you know, some of the examples you've seen of other founders, but I want to stick to sort of that zero-to-one stage, where, you know, the framework makes a lot of sense. But for founders to be brutally honest with themselves is critical.

Because otherwise, you know, the document is can be, you know, just a bunch of platitudes or mumbo jumbo. So, what advice do you have for founders who are evaluating these markets? My advice would be that if you don't believe you can be brutally honest on yourself, then ask others to look at your hypothesis document. That's actually one of the other values of having a written hypothesis document. Because then you can share it with your advisors, your well-wishers, your VCs, or potential VCs, and they can be brutally honest for you.

Right? And the worst thing that can happen to a company is if the customers out there are brutally honest on you after you've built the product. Yeah, that's a tough place to be. Right? So, so you might as well have if you can't be brutally honest, intellectually honest with yourself, then show the hypothesis document to people who will do that for you. And believe me, feedback is a gift. These guys are actually doing you a favor.

They're, you know, preventing tons and tons of years of frustration later on by being honest, right? So, so thank them for that. Um that's my advice to founders. And how do you balance the on one hand every founder has to be incredibly thick-skinned and be willing to take a lot of disbelief. Because every new idea is disruptive and people will say, "I don't think this is going to work. I don't think that's going to work." Uh and I'm sure you heard that from a lot of people and every founder hears a lot of that.

Skepticism. And if you if you respond if if you if you get beaten up by every skeptical sort of piece of feedback, you'll never start a company. On the other hand, as you rightly said, you do need third-party feedback both positive and negative uh to process ideas. So, how do you find that balance? Yeah, so it's first of all, write down all the naysayer comments as your why nots. Write down your rebuttals. Now, you may be swimming in Kool-Aid air uh when you write those rebuttals. But, the important thing is after you've written down those rebuttals, show the hypothesis document to a statistically significant number of people. Maybe five, you know, objectively smart people.

And if the bulk of them are saying, "Dude, this makes sense." then it makes sense. Don't just decide by yourself that, you know, I have good rebuttals and therefore it makes sense. So, it's an iterative process. You're you're getting feedback, you know, synthe- listen to it because the process of writing the feedback down means you have to synthesize it. You really have to understand what would be what are the naysayers saying. I'm also going towards a disagree and commit kind of philosophy where um you know, think about it as don't you shouldn't just be the um alone wolf.

That hey, uh this makes sense because I say so. If enough people out there are saying it makes sense, well, then it's probably true. Yeah. And if enough other people are saying, "Dude, this hypothesis document this doesn't your rebuttal it make sense to me. I really have concern here. Um then you should take that seriously. Absolutely. You know, another one of your checklist items which I think you said this well and I've experienced this very often is that founders often don't spend enough time sizing up the TAM.

And some of the smartest folks in their rush to get started go after small markets. And you can change product, you can change MVP, changing markets is not impossible but much much harder. What advice you have for founders who are looking to make this sizing decision? Yeah, the very first advice I have, Ashu, is actually to be honest about the TAM itself. So some So I'll give you an example. I Somebody came to me for advice on doing a company and they were doing a company in like the desktop desktop space. Yeah. They added When when I asked what is the the TAM, they were adding up the the revenue of Microsoft plus Intel plus God knows everyone who's working in the desktop space as their TAM.

And that's not the TAM. They They are not thinking about it the right way. The right way to think about it is that if you if that company was wildly successful in what they did and they killed all competition in what they did then the revenue they'll be making is their TAM. So they likely are not even building an operating system. Right? So they likely are not killing Microsoft. They likely are not building like a processor or that processor. So they likely The The The revenue that Intel makes is not going to come to them if they start winning in the market, right? So when you win the market, you have to ask what revenue is coming to you.

And if you are so wildly successful that every revenue of that sort comes to you and doesn't go to your competitors then the revenue you'll be making is your TAM. So you got to be honest about that. That's your real TAM and that TAM ought to be big. In some sense, it's a bottoms-up calculation to say in the fullness of time, what is the price per user you can charge? What are the number of users if it's a user-based pricing? Sometimes it's consumption-based pricing like different units. What's the unit of pricing?

What's the price per unit? How many units can the market have over time? Doesn't have to be today, but it can't also be 100 years later. Maybe in 5 years or 10 years. And when you multiply those two numbers and are really practical about how reasonable those numbers are, that's the tap. Like that bottom-up number. And really getting crisp about those numbers. And being realistic about what those can be over time. And and you can um get hints from if there is a similar product, maybe it's a legacy product, but it's similar.

You can take hints from that, you know, how much they're charging. So, maybe it's a legacy product, maybe it's

expensive uh and maybe your uh innovation is to actually cut the cost down by let's say 50%. Um so, whatever that company's making and whatever number of customers they have, right? Uh you can sort of assume that you'll have probably that many customers given enough time. And since you're going to cut the cost by half, well, you know, you'll make half the revenue. And so, you can do this for a number of your competitors, add it all up, and that's probably your your TAM. The point is if you have a big enough TAM, then you can get to 5 10% and still it's meaningful. Exactly. But if it's a small TAM and you get to 5 10% of a small TAM, you'll have a very hard time going up from that.

Any other examples that, you know, you can share without necessarily naming the founders or companies where, you know, you had to go through this process of getting people to be brutally honest about themselves or, you know, where people got into a confirmation bias or other common traps like that? Yeah, there's you know, without going into names again, there was a bunch of founders who were doing analytics companies. Uh I think some Some actually even came to you. I don't think you funded them, but uh uh they were doing analytics companies and they were actually adding the times of every analytics player in the market.

It's like this is my time. I'm like you're not even going to displace them. You're not going to not even, you know, going against let's say, I don't know, SAP or or Oracle or what have you, whoever else is doing analytics. So how can you take the time the whole market and say this is your time, right? You ought to be very honest about what your time is, right? And so that was the and sure enough the company they were projecting their time as very large.

You know, one thing about, you know, venture capitalist is that if you don't make a convincing case for your time behind the scenes somebody else is probably going to make an even conservative call, much more conservative than what the reality is and then based on that they'll quietly decide not to go forward. That's kind of what happened with some of these founders when they made, you know, not so great assumptions about their time. And they went belly up. Now some people they did get funded and then then went through pain. It's actually I I I think it's a blessing for people not to get funded when they're doing this kind of stuff rather than they do get funded. They actually do intend tend to convince someone somewhere to give them the money and then they go through four six years of hell and then they see, "Oh my god, I didn't realize, you know, my six years of my life has gone to waste.

Why am I so unlucky? You know, why are so many other companies successful? Why is it It's like then they're going to, you know, self-flagging and self-pitying and whatnot. It's just I tell people that look, take the time to find a high quality investor, whoever that might be. Yes. If you're not able to find high quality investors, it's possible you still have a great idea. So I wouldn't rule it out, but there is a lot of signal in that.

Surround yourself with great people, high quality investors, high quality advisors and they will tell you the reality, right? If you're off, they will tell you. But if you keep you know, drinking your own Kool-Aid and you're like, "These guys are wrong. I will show them." Okay, I mean yeah, sometimes it happens that people are wrong, but how likely that they're good people, they're all wrong, and you're the only one right? No, I I think that's very well said. Do Do one of the thing, Mohit, I want to press on is, because you didn't talk about it is very

often, even in large markets, you have to bring something that is truly innovative. So, how do you think about this notion of being disruptive as against incremental in in your MVP?

Yeah, so the way I think about it is that let's again think of it from the perspective of a customer. So, okay, the customer is going through some pain, and you're coming there to say that you're going to address that pain. But, please understand that making the customer change is also a pain. And if the pain you're solving isn't significantly bigger than the pain the customer is going to go through to change, the customer is not going to do it.

So, you need to make the customer go over the hump of that pain to actually change. And for that, your thing better be significantly disruptive. Another reason is that let's let's say your improvement is only 5%. Well, whoever that person who whoever that vendor is that the customer already has in their environment, when they look at somebody who's just 5% better replacing them, they're going to come to that customer, and they're going to say, "Dude, wait. We're going to give you these improvements in the next 3 months."

Yeah. Right? So, you better be uh significantly disruptive, so they they basically have a hard time catching up. Right? Otherwise, there's no point. Otherwise, if it just takes a few months for them to catch up to whatever 5% increment that you're doing, then there's what's the point? Mohit, before we switch gears to talking about product market fit, which is the second step, I'm curious if there are any other examples uh either in the positive or negative of founder stories at the very initial starting point that you would like to share.

I think I look I have I all heard stories about let's take service now. I mean if you read you know books by Frank Slootman there's a book amp it up. Yep. He says you know getting to the product market fit was a minor speed bump in the service now's history. The product was so disruptive and so attractive you know whatever pain it addressed was the pain was so significant that you know that that's a very positive story. So there was a real pain I think the founder of service now he saw that IT was a pain. Mhm. Every time you file an IT ticket you had to deal with you know IT people and you know he wanted to build something that he would eliminate that. And it was a real pain.

So that's a great story and there's plenty of other great stories right? Any any company that's a big hit that let's take Google as an example right? I remember that when I was you know graduating when I was doing my PhD how difficult it was to find anything on the web. Like there were like some 10 search engines and you go anywhere they'll throw a bunch of ads at you. The ads were very flashy bad ads and yet the information that you were looking for could not be found.

And here comes a very clean interface from Google and you find stuff it was so relevant that the top search results were right like contained what you were looking for. Yep. And so right there was very you know relevant pain that they addressed and sure enough it was a big hit it was a huge market which initially people did not see because any of the other search engine companies were using a business model that was not very big. I think Google eventually found a great business model through showing ads on the side.

That made it very very big right? But the pain was there. When they combined the pain with a large market, that's when magic started happening. If Google had put its mind much earlier into building like a maybe a hypothesis document, one of the why nots would be that it's a small market and maybe they would have found this bigger market much earlier on. I will tell you that I when I was you know doing my PhD, I was approached by Google to join them as soon as I graduated and I pushed back because I didn't feel the the market was there. It was only I think in the early 2000s when they discovered this ad market and then it became a big company. Right. The same story I think I believe VMware had the same story. I think initially they were only looking at desktop virtualization and immediately as they went into server virtualization is when the the big hit happened. But you know look these are these companies got immensely lucky.

They started small, they found the market and then hit it. I my playbook is to try and find the big market up front so you don't you know languishing around and you know trying this and trying that and the sometimes companies don't survive that. Okay. So Mohit, you know given your experience with Nutanix, Cohesity, having advised and invested in so many companies over the last two decades, uh if you fast forward to today, a lot has changed in the last couple of years whether it's the pandemic, the GenAI, you know, hype cycle, uh if you were if you were to do it again, is there anything you would change in your framework? How is your thinking evolved? Yeah, look one of the why nots I would like I would right. So even if everything else is solid, I think you are referring to what exactly is your solution?

Is it a good enough solution? So so if I'm solving a problem, but I'm solving it the old way, the danger is that somebody else would come and solve it the new way. And then they would be a much better hit. So I'm let's say, you know, let's talk about generative AI. I think it's definitely taken the world by storm. In so many places, um if someone finds a problem but they try to solve it without thinking about AI and there is a way to solve that problem by bringing AI or generative AI into the mix, then you're really missing out. Because once you made the world aware of that problem and once you made the world aware that this pain needs to be solved and you're allowing someone else to do it in a way that's more next-gen, somebody's going to do it. Yep. And then your solution would be toast.

Are you using the latest and greatest technologies to solve the problem? Right? Or are you using technologies from 10-plus years back? Because if you're using 10-plus year-old back technology to solve a problem, okay, you may solve the problem, but then somebody else will come and up-level you. And I think the thing I would add to what you're saying is when there is such a profound market shift shift happening, you sort of have to play that movie out and say the existing players, while you are better than what they're doing, how are they or others likely to sort of ride this new technology wave?

Because entire categories may become less relevant and new categories which are small today may become 10x more relevant. And so that you know, when you have this literally once-in-a-decade market shift happening. I mean, we had mobile was the last one and and mobile in the grand scheme of things didn't really change enterprise software all that much. It did, but was not transformational. The internet, of course, and cloud completely changed enterprise software. Uh and this shift is almost as big, if not bigger.

That's right. So, I think you're right. Uh if you were to ask the question that, "Hey, if a competitor comes and they have a mobile app for enterprise software and you don't, is it a showstopper?" And I would say in most cases, maybe not. Maybe in some cases it is. But generative AI is so transformational that you You I'm almost hesitant to say that if somebody building a next gen company doesn't have some aspect of AI in it, you're probably missing out.

Absolutely. I'm going to shift gears Mohit from this, you know, the the starting the first steps in the journey to really the the second big step is, you know, finding product market fit. And you know, even the word product market fit means different things to different people and people often talk about it in very hand-wavy terms. How would you define product market fit for an enterprise startup? My definition is very crisp. It is product market fit happens when an average sales guy is able to sell to an average customer without involving people in the headquarters.

Now, let me explain that. Right? If it's a very elite sales guy doing the sale, that is not repeatable. You can't be able to sell only through very elite sales people. Yeah. You need to be able to hire somewhat average sales people and be able to sell. And on the other side, your customers also can't be too elite. Yeah, elite customers might understand your whiz-bang product, but is an average customer able to understand it?

Right? So, an average sales guy is selling to an average customer. And then if the people in the headquarters, and by that I mean the initial team, right? Founders and whatnot. If they have to get involved in every sale, well, they're a limited commodity. That that's not It's not repeatable if they have to get involved to to sell the product. So, so product market fit happens if an average sales guy sells to an average customer without involving people like the founders and other elite people in the headquarters, your C-staff or what have you. That's a product market fit. Because because product market fit is all about repeatability. I think it makes a ton of sense and it you know, it's something that you've said many times and it's really always resonated for me.

And the revenue number will vary depending on the company because in some cases if you're if you're selling something for $100 a user versus a million dollars a user, obviously, you know, the revenue number will be different. Do you have a rule of thumb on how many transactions or customers are required to get there? Uh it depends on enterprise companies to uh so that, you know, roughly enterprise companies, you every sale is a uh is a is a high price sale Yeah.

Yeah. whereas there are these other companies which maybe every sale is a is a low volume sale. Like Google is an example. Yeah. Right? So, Google, the product market fit probably happens uh for a company like Google if it has a few million customers. Yeah. Whereas for a big enterprise company which is selling, you know, for high price, the product market fit potentially happens if it has, I don't know, uh 500 customers or something like that.

The numbers need to be statistically significant. Right? 10 is not a significant number. But once you reach more than 100, you reach something like 500 customers, you know that you've pretty much seen it all. Yeah. Right? And that's when product market fit happens. And as you, you know, as you go back to the the stories, you know,

your experience at both Nutanix and Cohesity, can you share some stories of you know, how did you just feel it? How did you know when you had hit product market fit? Was there an aha moment?

Yeah, yes, there was. So, Um Look, the fou- I was the founder and CEO for the longest time, uh 9 years at Cohesity. And the biggest deals, look, more often than not I'm I am involved uh with the biggest deals. Yeah. Uh but then there were these nevertheless big deals that started happening where I was not even involved, like not at all, like zero. And that's when I knew we've hit the product market fit. That my team can close it without me.

Now they'll try to up the odds by still involving involving me in some very important ones, but when like the Fortune 500 deals started coming and I was I hadn't even spoken to the customer. That that was it. That that was my signal that we would achieve product market fit and you know the company actually needs to operate very differently once it has achieved product market fit. So there's there's almost like a a line in the sand so to say. Before you reach product market fit, you're conserving cash.

Right? You're doing what it takes to get to the product market fit, not expand the sales force or what have you because heck, if you expand the sales force by a ton and every deal requires the founder, well, they can't sell. Right? You're just going to burn money. But once you reach product market fit and the definition again is that, you know, average sales people can sell to average customers without involving, you know, people in the headquarters, then it's all about pressing on the gas.

And it's a mistake to not press on the gas, right? Otherwise, you're just opening up to somebody else taking the market from you. Because people know now that this thing sells. There is a pain here, this thing sells, they're probably going to build something like this, they're trying to copy. And if you are too slow, they will, you know, get the market that you're supposed to get. You know, I think you've hit on something really critical and something I always struggle with. If you flip the switch too quickly, then you start churning through sales people.

Because, you know, the first three or four sales people or five sales people worked really well because really it was the CEO selling and you're thinking that your sales people are selling, but you're showing up in enough of the meetings that they're running a process and suddenly, you know, if you say, "Look, I'm going to go from five to 15 or 20." You can't duplicate yourself and and you and you go from having very successful sales people to the model literally collapsing.

That's right. And and you know, you start burning money because sales people are not cheap. So every time you hire a sales guy and they're not productive, they're not selling, well, guess what? The company's paying for them. And not only are you churning sales people, you're actually burning money and you know, then your quarterly numbers start getting missed. You know, you were if you had if you doubled your sales team from one quarter to the next one, but your sales numbers are not growing.

At that point, your VCs are kind of asking, "What the hell? You know, what what happened to your financial plan?" So it becomes almost like, you know, a snowball effect and it almost becomes like you get choked from

multiple multiple angles. So I I don't know That's why I said it's literally a line that you shouldn't cross unless you're sure that you've reached product market fit and then you can grow. But until then, be careful. I'll give you stories of Nutanix when we hired sales teams and we expected, well, they'll probably not do anything the first quarter. And lo and behold, they did 500k the first quarter.

without involving us. What are some of the other lessons learned for you as you think about sort of product market fit, things that have worked for you, mistakes you have made or seen other people make? Yeah, one one big mistake and this tends to be a sensitive point because it involves people. You know, one thing the success of a company heavily heavily depends on the team that you hired. And it starts with your lieutenants. You know, if you're the founder CEO or C-level exec, your immediate team then, you know, it basically replicates below that.

And some people have such a hard time A, you know, hiring well. And if they make some mis-hires, they don't performance manage them out. And please also realize that somebody might have been hired might be great for a certain phase of the company. Two or three years later, they're no longer a fit. And now if you are just trying over hard to make it make it work, you're just not doing anyone a favor. Right? You're not doing your company a favor. You're not even doing that person a favor because they're probably in pain, too. Performance management is unfortunately one of the most important things that is needed as you are rapidly expanding and growing, especially as you reach product market fit. Remember before product market fit, you're probably not growing as much.

But, if once you reach product market fit and you're growing something like 100% like 100% means that in in 3 years, you're you're you've quadrupled. Right? 1 year, you doubled. The next year, you've doubled again. So, in 2 years, you've you've quadrupled. At that point, a person you may have hired 2 years back, is he or she scaling? We don't know. Yeah. Yeah, we don't want to make it a hire-and-fire culture, but after trying for one or two quarters, if they're not going to making it, you should replace them. And the challenge a founder faces is also, you know, sometimes from their board because the board probably has very accomplished people, but they probably came from more mature companies where that churn isn't there because the company isn't like maybe the company is growing at 10 15% and they're not churning people like that.

So, they would even have some pushback that, "Hey, why the heck are you, you know, firing people or letting go of people every 2 to 3 years?" So, that's something that a founder has to face. You know, high-growth company, it's just the way of life. You know, I I attended a fireside chat. I will not name the name, but very accomplished individual, and he said that the likelihood that your executive staff will last 3 years is very, very low.

Very, very low. The probability that a given person on the executive staff will actually survive beyond 3 years is rather low. Some will. It's just statistical. But, not many will. And if you just insist on not changing, you know, you will suffer. Your company will suffer. I think well said, and part of the challenge is that it's also a small small sample size problem. So, if just to pick a number statistically, 20% will survive, there's a distribution around that. So, it's entirely possible that you fall on the left side of that distribution, and none of them will survive.

I would say if you're intellectually honest, you know, it's the founder's job to say, "Hey, I want to step aside." But, sometimes the board has to make that call. That look, this guy is not scaling. You know, that this person I was talking about, uh he has he gave a very good litmus test for uh checking if someone needs to be replaced. And that litmus test is that if the team beneath you is not strong for that stage of the company, that is a litmus test to say you need to be changed.

So, so if the founder, CEO, whoever, if the team beneath that person is not strong, it's not the team that needs to be changed. It's the CEO that needs to be changed. And sometimes that is where that is the hard part that where it gets to. But, the way for a CEO to scale, as you rightly said, is to ensure that they are scaling the people below beneath them. And sometimes it means pulling the band-aid. You know, the other thing I wanted to transition to is you know, one of the things you've often talked about is is really as a technical founder learning go-to-market and learning how to build and scale go-to-market like a science. I'd love to spend a few minutes on that on on best practices, lessons, mistakes that you have made that you would like to share.

Yeah, so the first I would start is hiring. Um people just like this is the way people hire, right? They'll just go into an interview and yay or nay. The problem is that that approach might actually work for, you know, maybe engineers, and you can ask them questions on the whiteboard and, you know, make them solve some puzzles. But by and large, if you just want to have a discussion with someone, uh it's probably not going to work.

They're probably at their best behavior. It's uh if it's a business role, and I myself was a victim of that, right? I would literally I found that as I reflected, I was pretty much hiring people because of a chemistry match, not because they were capable. Right? So, it was a hit and a miss. And that's when I, you know, came up with like this I would I call hi- competency-based hiring strategy, where I divided up their competencies or what I desired in that role into three checklists or scorecards, if you may.

So, the first one is what do I look for in a resume? I don't want to waste like tons of people's time if their resume is not a match for what I'm looking for. If I don't have a checklist, I'll probably miss it. So, for instance, let's say I'm hiring a VP of sales. Right? I probably what might be important to me one of the checklist item is that they're coming from a big enterprise company. They may have They have 5 plus years of sales leadership experience.

Right? They're coming from reputed companies. Right? These could all be checklist items that you could just match against the resume. Why waste time interviewing them if they don't even match that basic stuff? So, once they match that, so I call it the pre-interview scorecard, then we move on to the next step, uh which is interviews. And in the interviews, you again have a checklist. Right? So, one of the checklist items there could be um you know, I don't know if is this guy good at hiring and performance management?

Another one is does he Is he really good at sales methodology? Does he observe the right kind of discipline and all that stuff? So, there could be a number of checklist items there. And um no one person can test for all of them, right? If you have the person interviewing with, let's say, six or seven people, uh the idea is to give uh two or three checklist items to every interviewer. Hey, you test this. I will test this.

And there may be some overlap. Uh no one person should even make an attempt to test all of them. Right? And and finally, there is a third scorecard that I use, which is for reference checking, right? So, for instance, uh if if I'm calling a reference and I'm asking, "Hey, is this a good guy?" That's like the absolute wrong kind of reference calling. Because probably the answer that will come from the other end is, "Yeah, he's a great guy. Go ahead." That doesn't tell me anything.

So, what do you want to ask is, you know, some of the possible questions that I write down in checklist format, is, "On a scale of 1 to 10, would you hire this person again?" So, if they say six, it's a fail. If they say eight, I would say, "Okay, why is it not a nine?" And why is it not a seven? Give me some examples of why you think it should not be a nine. That brings out the problems.

Right? Uh another question could be, you know, "Did high performers at your organization respect this person?" You can validate some of the other stuff. It Is he good at hiring and performance management? Would great people follow this person? Is he a good culture builder? Right? And you ask him all I like to ask him on a scale of 1 to 10 because when people are sugarcoating, right? If you don't put a scale of 1 to 10, they'll say, "Yeah, he's a great culture builder." But you ask him, "Hey, on a scale of 1 to 10, how good of a culture builder are they?"

And they will maybe say seven. I was like, "Okay, that's actually, you know, somewhat average score." No, I I I think what you're saying is very interesting because the the 1 to 10 scale is truly actually a 6 to 10 scale. You know, anything below five, six, seven is all in the negative category, but it gives people an out. That's right. To be you know, to provide faint praise, so to speak. Yeah. So so anyhow, your hiring ought to be really good, and you need to definitely do some blind reference checks. The biggest mistake one big mistake people do is okay, they ask the candidate, "Can you give me some some references?" Of course, the candidate is going to give them, you know, their friends or what have you, right? So you need to it's the onus is on you to go scour LinkedIn and figure out who else this person worked with, right? And call them up and get a reference check.

So the first thing is hiring, and it's it's really this notion of a a screening checklist, an interviewing checklist, a references checklist, and blind references. Then the next thing you need to, you know, track their progress. And one great way I have found to do that is through OKRs. Because otherwise, you know, it's all in the air. You expect, you know, them to do X, they do Y. There's no track quarter after quarter, you know, unless it's like a real C player, you know, people kind of in the middle will keep bullshitting you.

So whatever you expect them to deliver, just put them in their OKR. Right? The OKR should be you know, as you know, it's objectives and key results. It's a I I found it a great way to you know, direct what a what a certain person ought to be doing. So you go to of course with each of of your left lieutenants, you negotiate with them what they're going to deliver on. And at the end of the quarter, you both agreed that so and so things need to be done, and then you rate it that okay, we don't expect the score to be 100%, but I hope the score is closer to 80%.

Right? But if it's like a 50%, then we got to question why. I mean, did you have too much on your plate or you're just weak? Right? And if it happens to be 50% over and over again, then you know, you have a performance

problem at your hands. And that brings me to the last point, which is performance management. Right? So, here's the mistake people make or a couple of mistakes people make on performance management. One is that they will only have Well, they will not even have performance reviews. That's like one big mistake. Yeah. Second big mistake is they will only have isolated reviews.

And that's it. Well, not having performance management is of course bad, but also having like a 360 where individual people just write down what they think because, you know, people are very shy of writing down the reality on paper. The right thing to do and you know, I I love doing calibrations. Yeah. Which is really a meeting where people come together to discuss the performance of a person. And that's where the real stuff starts coming out.

Right? Some person might be Is it only me who's thinking this way? But then three other people are saying the same thing. Then it all comes out. And that's where So, so you can do a 360 with individual reviewers and you can give an initial rating, but that rating needs to be updated in a calibration meeting. And that's when the real rating get comes out. And my rule is that I use a five a five-point scale.

Um so, the lowest part is basically the person is not meeting expectations. Right? You can call it a C player. Mhm. The one above that is the person is a is is sometimes meeting expectations. So, he's a B player. The one above that is you know, the person meets expectations. So, he's a B plus player. The fourth level is the person exceeds expectations. So, he's a A player. And the final one, the fifth one is significantly exceeds expectations or A plus player.

I want anyone B or below to be out of the company if they can't move up. And you can put a deadline. Maybe it's one quarter, maybe two, but definitely not more than two. Because I think Frank Luntz also says the B players are like the ultimate passengers. They are the ones who will kill the company. The C players you'll get rid of, you know, sooner or later. But B players are the ones who probably are doing something right.

But you won't hire them again. That's my litmus test for a B player. And you want to get them out because otherwise it'll be death by a thousand cuts. How do you deal with sort of, you know, And and you know, Frank talks about it a lot in Ambitup as well. Uh that our culture where you are systematically sort of identifying and managing out B players. Uh you know, keeps everyone on their toes. It does It has cultural implications.

Uh some good, some bad. And it breaks some glass. What has been your experience with dealing with that? Yeah, I I'll say two things, Ashu, on that. One is you make it HR's responsibility. Your HR should just do this like clockwork. Right? You just have, you know, uh either quarterly or um every other quarter these performance calibration meetings. And we're not saying that it's a hire-and-fire culture, but you get flagged as a B or a C player and you're basically on a performance plan. And the intent is to actually up-level you, but if if you cannot be up-leveled, then you're out.

But even more importantly, it's setting expectations that guys, this is how we operate. If you don't agree with this, just don't join us. Don't surprise people. Right? Uh make sure they understand that this is the culture. It's a

high-performance culture. Uh people who are doing well have nothing to fear. Um people who are not doing well will be told where they stand. It's a very transparent culture. As opposed to a person doesn't even know they're doing well or not and then one fine day you're like, well, you're out of the company.

So, people actually appreciate knowing where they stand. And by the way, good players thrive in this environment. In in in a framework like this, you know, once you have enough scale, obviously in small numbers you can't do anything that is statistical, uh how do you think about having absolute performance expectations as against trying to put people on a curve? Yeah, I I can't don't believe putting people on a curve. I mean, maybe you can put it on a curve from a bonus perspective, but not from a firing perspective. Like, imagine you have a great team, all of them are great players, but you are forced to put them on a curve and mark one of them like a B- minus player. That just doesn't work.

If they're all up, you know, good, then just call them all to be good. Now, there can be shades of good, that's fine. So, you can decide your bonuses that way, but don't mark them a B player just because you have to mark a few people as B players, right? So, keep it absolute and then have the culture force managers to be intellectually honest. That's right. So, because if they're not, then the managers should be flagged as, you know, not a B plus player and they should be pushed out. If you went back to the earliest of founding Cohesity, you know, if you had a time machine, you could go back almost 10 years, uh what advice would you give yourself?

What do you know now that you wish you had known then about building an enterprise startup? Yeah, I I was a I was a technologist before I started Cohesity. And I would hire everyone, um you know, I would interview everyone like a technologist. And that was one and I sure enough I made enough mistakes, especially on the business front. I hired a, you know, very some very bad hires uh because I did not know how to hire.

So, the first advice I would give to myself is be humble. You actually don't know a a lot beyond technology. Right? You have to learn a lot. Right? Uh and you need to learn how to hire business people. You need to learn how to build high-performance processes in the company. The sec- the second thing I would say to myself is I need to start learning about sales, especially the science behind sales. How do you inspect sales people?

How do you inspect go-to-market? Whether go-to-market is doing well or not. I mean, there's a uh you know, Kal Ramanathan was uh was was a board member of ours and he said something, uh you know, very illuminating once. He's like, "Uh you can talk about all sorts of innovation. There's product innovation, this and that, but there's one innovation that trounces every other kind of innovation, and that is GTM innovation." So, unless you understand GTM, and most technical founders don't, Yeah. somebody would probably build an inferior product, but a superior go-to-market, and they will trounce you.

So, start learning about some of that stuff, because if you're a technologist, the last rest of your life, you probably don't know that stuff. Right? Start There's some great books you can start reading uh to get some knowledge. And some of it comes from experience, of course, but at least you can get a head start. Don't be foolish enough to think that just because you built a great product, it'll just sell. Absolutely. Founders who are in the ideation of pre-product phase, so even before they've started the company, and you know, if you had to pick one insight,

and we've talked about a lot of things, uh but if there's one thing you would want to leave them with as they embark on this journey of starting a company that you've now been on so many times, what would that one thing be?

Yeah, I would say, you know, have a a bulletproof hypothesis document. Keep updating that document as you I mean, no matter how good you are, you know, the hypothesis document, you probably have limited uh touchpoints before you start the company. So, you will discover more about your product as time goes along. So, keep updating that, and make sure that you're always kosher. The the why nots, you always have a great response for the why nots. Right? Uh otherwise, you probably need to enhance the product Uh some ways so that you can address the why not. So always have a hypothesis document that is updated in real time. I would definitely advise that.

And I would say one one more advice I will give to people is that eventually repeatability comes from processes. Right, so systematize stuff. So I I like to give this analogy. It's the same people I show you and I have both grew up in India. Yeah. It's the same people who drive badly in India and and come here to the US and they drive well. Yeah. What's different? It's It's clearly not the people. It's the system.

It's the traffic system. So if you build the right systems in your company, the desirable outcomes happen by themselves. So if you build the right hiring system in the company, your hiring will get better. If you build the right performance management system in the company, your performance management will get better. If you build the right sales systems in the company, your sales will get better. So think systems. Don't think individual people. Exactly. No, I I think I think that's very well said.

Uh Mohit, I'm going to switch gears to another topic now and you know switch away from your experience looking back and sort of you know, look forward again and say, you know, we talked briefly about generative AI. Uh it's a very unique time in technology broadly. It's a unique you know, we you and I both saw this early in our careers. This is the last time we saw something this this this disruptive with the internet and the start of the cloud.

Given what's going on in the world around us, you know, what are you most excited about? What are the What are the big things in enterprise technology that really get you feeling like, hey, this is why the world will be different over the next three to five years. Yeah, so what's exciting Ashu I must say is that given uh you know, what kind of the given the kind and magnitude of disruption that generative AI can can bring pretty much any area can be disrupted if you just think hard enough.

Take any big company that is doing quite well, but that's not using generative AI can possibly be disrupted. So, that's the exciting part. Right? So, if there are founders out there looking to do new ideas, not only should you think about yeah ideas that go beyond generative AI that don't even involve generative AI, but try to marry to generative AI and you may be able to bring a huge disruption. Right? So, that's very exciting that pretty much anything you can think of from any anywhere from a HR company to a insurance company, right? You marry generative AI to it and you can think how disruptive it can be. You know, as we were talking about sort of you

know, making the transition from being a technical co-founder to being a CEO and running learning all the business aspects are there one or two resources, books, podcasts, newsletters that you would recommend to other founders?

Yeah, there's look there's lots of great material. I I will do injustice if I only recommend a few, but I will only recommend a few. So, I think Frank Slootman book I already mentioned Amp It Up, that's a great book, phenomenal book. To learn sales, there's another phenomenal book from the author John McMahon, you know, the book's name is The Qualified Sales Leader. Right? That I think people should see. To to hire well, there's a book called Who.

It's a book on hiring. Right? There's a great book from Amazon, I think it's called Working Backwards. Um absolutely phenomenal, absolutely phenomenal. It touches hiring, it touches metrics, it touches so much more and I can now see why Amazon is where it is today. Right? I I think these are some and there's a I think in terms of leadership, uh, there's tons of great books, but I think of this one, Leadership and Self-Deception. Uh, that's a that's a great book. So, I I would say these are some great books that someone can start with.

You clearly read a lot. I mean, that in in itself is, you know, five great books. And maybe the last parting question, given how much disruption there is, there are people who are There's lots of naysayers about technology, lots of people who talk about the doomsday scenarios. Uh, but, you know, in that in that ecosystem, like what has you most excited? I I should say, you know, I'll just tell you a very humbling story first. Uh, I remember in the in the in the late '80s, I mean, I was I was in high school, and I remember uh, going to meet someone at the time. We went to Delhi and we we met some friends and somebody came who was involved in the early computer companies. The person, uh, you know, came from the US and he was from one of the premier schools in India, the IITs.

And, um, you know, I remember us asking that person that look, you already in the world has already invented computers. What's more to invent? What do you guys work on than computers? Right? So, So, I think as we look back, you know, you can only, you know, look at it a little bit humbly that, you know, you can't even predict uh, up front what's coming. And that's really what excites me, that I actually don't I'm I I I I would not be acting humble if I were to actually predict what's coming because the future is just just the so many possibilities.

But, yeah, immediately I see lots of possibilities that AI can bring. I'm very excited about the some of the stuff going on in let's say biotechnology, right? Some of the compute power uh GPUs and whatnots are now, you know, uh being applied to uh to to nanotechnology, to biotechnology, to biology. And what that might do and, you know, all the great stuff happening with GPUs and very large uh compute capacity is available, uh web farms available. Um imagine the CPU power. Imagine the computations that can be done on, you know, your genome sequences. And it's just like fascinating, fascinating where people can take this.

We're all set for an exciting decade ahead. Well, thank you so much for joining me today, Mohit, and really appreciate all the all the time you spent on this. Absolutely, Ashu. It was fun. Thanks a lot. Thank you.