

AI Doesn't Replace Builders. It Amplifies Them.

55 MIN · YOUTUBE · [HTTPS://M.YOUTUBE.COM/WATCH?V=SNCT0XJAAVO](https://m.youtube.com/watch?v=snct0XjAAvo)

<https://m.youtube.com/watch?v=snct0XjAAvo>

SUMMARY

The discussion revolves around the challenges and transformations in software development and startup culture, emphasizing the importance of adaptability, team dynamics, and the evolving role of technology, particularly AI. The speaker shares insights from their experiences in engineering leadership and venture capital, highlighting the need for a blend of generalist and specialist skills in today's tech landscape.

- Starting a project may seem easy, but scaling and complexity present significant challenges that require experience.*
- The lines between product, design, and engineering are blurring, necessitating a more integrated approach in teams.*
- Emphasizing the importance of understanding customer retention over acquisition, as addressing churn can save resources.*
- The speaker advocates for empowering teams with autonomy and trust, fostering a culture of responsibility and transparency.*
- The evolving tech landscape requires a shift from traditional roles to more versatile positions that combine product sense with technical skills.*
- The speaker is currently focused on building tools to help non-technical users leverage AI for mental health support and other utilities.*
- Personal growth and self-discovery are emphasized, with a focus on patience and acceptance in both personal and professional life.*
- Recommended readings include "Philosophy of Software Design" and insights from the Gita, which provide valuable perspectives on duty and behavior.*

What's going to be the first block they going to hit? >> It's like Ferrari. Starting a Ferrari is easy. Driving it on the roads is tough. So, white coding is like a Ferrari. So, you are good when you are empty field, you're running like a spinning like anything. >> Wow, I just generated so many lines of code. >> >> And then >> So, why doesn't it work on the server, right? >> And since you have not done many other things So, now you will start and it will work for first few stages even for servers, deployment, everything is fantastic. And then you'll start hitting the roadblocks of a scale and complexity.

So, the code complexity goes up up. To match the scale, you will have to have microservices or you have to have event-driven architecture, you would have queues, all that is stuff comes only from experience. That's where they will hit. But, for first 80 to 95% they're very happy. >> >> AJ, so excited to have you here. We have >> Same here. >> such a >> Thank you for inviting me. You guys are so generous. >> >> For me, bringing this type of person is contributing a lot to the ecosystem. You have so much knowledge, you saw so many things. And whatever you going to say, I think it's going to be priceless even though you going to be modest and you don't

going to think so.

That's what I'm thinking. >> Yeah, I think it'll be interesting cuz you're a marker for me on my whole entry into what I call the Southeast Asia curve. I know it's been up and down of things kicking off with bonds and >> again. It's happening again. >> Let's hope so. Let's hope so. You've done so many things, right? But, I wanted to hit you with a hard question right from the very beginning cuz I see you out there coding away with Jevon and Claude and you're hooked like everybody else. I think if you were starting from scratch today building not saying a solo company, but building let's say a startup, funded startup or something and you had to build a product and engineering org cuz that's your thing. What would it look like today to you that must look totally different from how you would have done it years ago?

>> Well, there are two things. One, if I am the guy who has done something in past and now doing it now, then it's very, very different. For example, the product engineering role will expand. The lines between product, design, and tech will engineering product design EPD we call it will blur like amazingly in terms of going in somewhere where you don't have those differentiation. And then the role the ratio will flip.

So, you would have more people with product sense and less people who are building it. Because your builders are virtual now. All right. So, given that So, that is one one side of it. >> Being there an agent. >> There are agents, yeah. There are agents who are building it. We are supervising it and that's what we are doing right now. When you saw me outside, I was like wipe coding it or whatever you call it instruction coding it, but I'm trying to make sure that two agents talk to each other. I verify things, but I still look at the code.

But if I would have not done what I have done in past and I do this, then I would do which everybody's trying to do. Just add more engineers and try to build something out. That is also fine, but they will hit the roadblock at some point of time because they have not done enough analysis. >> What's going to be the first roadblock they're going to hit? >> So, one of the things which people will hit which I can see once product See, it's a it's like Ferrari.

Starting a Ferrari is easy. Driving it on the roads is tough. So, wipe coding is like a Ferrari. So, you are good when you are empty field, you're running like spinning like anything and >> Wow, I just generated so many lines of code. >> And then >> Oh why doesn't it work on the server, right? >> And since you have not done many other things So now you will start getting and it will work for first few stages for even for servers deployment everything is fantastic. And then you'll start hitting the roadblocks of a scale and complexity.

See the code complexity goes up up. To scale to match the scale you will have to have microservices or you have to have even driven this architecture you would have queues all of this stuff comes only from experience. That's where they will hit but for first 80 to 95% they are very happy. And I see that now. I see that now in my multiple arcs like when you go and say Oh our all our engineers are doing AI coding, right? But you saw recently the big meeting Amazon had and then the we need to have this review, we need to have this, we need to have that because people do things with guardrails will always go long way.

People who don't do things with guardrails will be always be watching for the swords to cut them thousand pieces. >> Are you referring to only for the engineers population or simple people? >> Everybody. Everybody if you look at design, if you look at product thinking. So the so LLMs can expand your PRDs to like 10 pages. And many times product managers don't read it. Unless they should read it. And if they read it they can make it tighter. Because LLMs can produce enormous amount of code and enormous amount of text.

And where we are right now and especially in last 6 weeks the text accuracy has gone I think it's gone 50%. So now people are not even reading it. So that is the problem because we will eventually have the problems of a scales which are very subtle in the beginning but drastic in the nature in the later life. >> But nobody has seen it yet. >> Do you think it's great like a world replacement meaning that the AI is actually teaching the human being instead of training the human being instead of vice versa?

>> That is happening. So, I'll tell you what what happened with me. >> >> So, it's so it's addictive. So, what happened was I and I tweeted about it and one of my ex-colleague also responded back. He also wrote the blog about it. So, what is happening now? Perfection has no end. Right? So, I wrote something saying do this and it did that. And now I'm like, "Hmm, this button over here this is should be a little left." So, you put a next prompt, make it a left. And then you are going on perfection spree where AI is driving you instead of you driving trying to drive AI because now you see everything is possible. CSS is no more nightmare.

Actually, CSS is no more nightmare for anybody now. >> Yeah, I think the problem I run into and I've had to change how I work is almost like what you just said. Like, "Okay, I deployed a new homepage. I look at it. Oh, I want to fix the button. I want to move this down. Oh, and this thing isn't very performant." And something And then I would tell them all of these things and I would start working on it. And then I would I inevitably break something cuz I shouldn't be doing >> Yeah.

>> four things at the same time even though it lets me. >> Yeah. >> And then I would find that I ship it and I broke something else. And then now what I realized is I got to stop doing that. I got to Okay, I have one problem and I'm going to talk to you about the the or feature. And then when I get that done, I'll go and move the button. >> Yes. >> But I don't work on the feature and the button and the color palette all at the same time.

>> because the AI become your your master splinter. Yeah? He's telling you what to do and sometimes it's just coaching you. Do that stupid mistakes >> Yeah. >> and you will be punished. Yeah? Until you're going to learn how to work with me, right? >> old world, we used to have something called yak shaving. Have you heard about yak shaving? >> Yes. But explain it. I like Okay. Yeah. >> So, Yark sharing is this very simple thing. So, I want to deploy a server or a software on server. So, I haven't deployed till now using say a script.

But now I need to deploy two boxes. So, I add the script, add boxes, but this is not good. Suddenly you think this is not good. I should do something else. Now, what should I do? So, I go, oh, I need something which can execute the script. Okay, what can execute the script? It can be CI tool it with Ansible. So, okay, now let's write Ansible script to provision the server and I put that in shell script and hook that at tool CI, but I don't have CI server. So, I now go and install CI server. While I was supposed to do only one thing, just add one more damn

host and move on.

>> Yeah. >> But now I'm like trying to add this CI server. So, I go to that CI server. But how would this my application boxes will pull from this? So, I need a registry server where I can store that artifact. So, it can get the build from both boxes. So, now I have one more task. Which creates registry, zip set it, uploads into some HTTP server, then it got. But how they will discover this? So, now I need service discovery as well.

>> And can you give me a dashboard so I can watch progress? >> You are just going on something like else and down the lines 7 hours later you have still not deployed. And it's called Yark sharing. >> Yeah. >> AI is trying to do that Yark sharing right now with perfection. You are going through perfection and since it is instead of you driving the AI, at some point of time LLMs are driving you. >> Just more tokens.

>> Yeah. >> Yeah. >> See, your background, you came from Rail Engine doing greeting cards, right? You've been a CTO or you've been a partner in in a VC. >> Operating partner. >> Operating partner. Sorry. >> Yes, that's the best part sorry. >> Operating partner. >> I put those higher up than partners. Is that just me? That's people that actually do something. >> >> You are the perfect guest for this show because you are a true generalist. What do you think that specialist missed being a generalist?

>> Okay. So, both a specialist and generalist, they have one thing in common, they want to build. They're builders. But the one thing is very different. If you look from my intuition, I wanted to become a real engine driver, as you said. That was a job, but then I got a printing press and started building business, printing business cards, and later on other things. What happens is you are always trying to you're always trying to scratch some itch.

It's about trying to do something which changes something in your life and for somebody else. You're trying to always scratch some itch. Over the period of time, when you look at this bunch of experiments as generalist, right? What generalist are good at? They actually start recognizing patterns and build the playbooks. What a specialist are good at? They actually know the playbook, and they have done that playbook again and again and again and again. So, when you look at a generalist, you may say that he cannot do this or she cannot do this.

But give them the tool and see what happens, right? Like for example, when we built Go-Jek team, none of our people the first 200 people, none of them had built any ride-hailing co- any ride-hailing platform ever. They were all generalist, and we actually made it very clear that we want generalist. Because generalist are so fluid that they don't have any, what do you say, they don't have any strong views.

Right? They don't have They don't have strong >> They don't have They don't have And I always say >> having strong opinions is great as long as you loosely hold them. >> Yeah. >> Correct? And that's what we had in our team. So, everybody was just a coder, or everybody was just a product manager or he was just a designer. Then you feed them the domain knowledge and see something coming out. So, that's what I think about generalist and specialist.

And later on you need specialist. For example, you need specialist who can only do cloud because your scale is so high. You need a specialist who can only do what you say backend engineering, only do mobile engineering, which is okay. But in the early days and even later days also, as generalist scope as expands as long as they can accommodate things and ready to learn new. Generalist actually have much much better role to play and shape the teams than a specialist do.

>> So, you mentioned Gojek and I think what would be fun if you have any old stories cuz I think when you kind of go from I'm in India, I'm at Koding, I get acquired, I'm eventually group CTO or CTO of this thing that I'm sure just kept getting bigger and bigger in front of your own eyes. There must be some funny stories in there somewhere about how you kept that thing together. >> Yeah. >> Because a lot of times what people don't realize, I think sometimes these things are only a moment away from catastrophic crashes or things not working, but you just don't see it, right?

>> I'll tell you this one thing. Think about this, a company growing >> >> 5% week on week. That in terms of traffic. In terms of traffic. >> Yeah, not month on month, week on week. >> Week on week. >> Yeah, week on week. >> Not month on month, but week on week. In early days So, when I reached there in 2015, once you started unlocking the Gojek was very unstable system because demand is so high and all systems are falling apart all the time. We started building things and then companies are still growing week on week, right? So, there's chaos all over the place.

So, now what is happening is we are going through bunch of rounds, we are investments going on. So, every investment round we get a bunch of people will come and do due diligence. And once they do due diligence, you have to go show up yourself, explain your system, and it was my day, so I got up at 10:00, get in the room. The room is full of this very large institutional investor. Room is full of all this lawyers and attorneys and like analyst and all this stuff, like some 30 people in black suits. And I'm like t-shirt and shorts and standing there explaining them the whole thing and like fine, everything is going well.

And I had half an hour slot, half an hour slot gets over. Just about when it's about to get over, I get a message saying system is down. >> >> And now since system is down and I knew I had to explain this amazing thing to them, like we have like we have >> We never go down. >> Now we have ghost and we go and we have show shown them dashboard, live dashboard and everything like this, like how beautiful it is. Everything is good.

And I get a message system is down. Like okay, fine. I have to stall them. And now I'm trying to stall them. And the funny story is that they don't know the system is down. I know the system is down. So I said, "Wait, I need to show you our architecture. We have very nice event source distributed system architecture and we have to show you." And then I go on explaining them the architecture for next 10 minutes.

And I see some of them are yawning. They are not interested. >> >> But they are they don't even know how to get around because the room has only one door, which is next to me. So I'm standing there, so they can't they go out, right? Finally, after 10 minutes, I get a message system is up. Okay, that was our architecture diagram, there's all this and they like they were sigh of relief. I had sigh of relief. They did not know what happened. I

knew what happened, but I saved the thing. It was a very amazing thing to do.

>> Total stall tactic. >> >> Like you should have seen when I was trying to stall them and talking all technical stuff, none of them were interested. >> Yeah. And it's okay. >> And you're just waiting for the message, we're back. >> And see it while you are telling this story, I'm thinking, hey, AJ have an amazing team behind him because you can speak with people while people are fixing stuff while you are not there. >> Oh, yeah. Yeah.

So, the thing is our team was amazing. We were like one big huge organism which is moving together as a one team. And one of the things which I always am very proud of is that what we built that point of time. You know what happens is if when I one thing which I did long back when I joined Gojek, I wrote two blogs. One is called what it takes to shine at Gojek. And Gojek engineering 101. And we put our principles down.

One of the things startups should do is to have a constitution. See, every every institution operates on some constitution. >> Whether it's stated or not, right? >> Whether it's stated or not. >> Hopefully. >> Yeah, they do. If you have If you have founders doing certain things, you'll have the next senior manager who has a close point start doing the same thing. So, that means you also need to know how do you think and how do you write that into writing that what do you think? How do you think?

If you can codify that, if you can put your thought process around that saying this is how we code, this is how we look at people, this is how we operate, then people will ask this question to themselves if Ajay is not there, what will happen? And they are empowered. And as long as they are empowered and not looking for your approval, you're good. So, we followed that. That means I had to do things which I'm expected to do like being hands-on.

We expected everybody to be hands-on. You can't be sitting in ivory tower. That the constitution or that the way you think if you can somehow communicate across, things will work. Second thing what we did, we actually made tech own the business metrics and business own the tech metrics. And we would send out our metrics report to everybody in the organization every day. How many orders we completed, how was our day look like, what graph line looks like, everything. Everyone.

>> Business or tech? >> Business or tech. And people were like shocked. You can't do this. And I said, "Why?" Because it will get leaked. Well, it's going to get leaked anyway. If you put a responsibility to engineers, product managers, or QAs, or designers, if you give them responsibility that this is the most precious thing and you are custodian of it, it will never leak. The problem is with lot of time we try to hide things, so people get more curious. You know, open, nobody cares about it. Even if you're like if you're like golden watch, if you leave it open it's like norm that we leave the watches over here.

>> Yeah. And by the way, this is thought through. This is what happened, I don't know, monday.com. They did the same thing. Everything was transparent on the monitors. The engagement, incredible, right? >> our our our dashboards were all over Gojek. >> Yeah. >> We used to have these dashboards and we would we would get obsessive over the dashboard. If line goes down, people everybody's like alert. What the hell happened? What the hell? >> Cuz they all would see it. Yeah.

>> Yeah. >> So, if you choose one action, because you mentioned like few things that you create, your team have their own autonomy. One thing that I'm going to take as a takeaway from that. >> I always say earn people and not money. If you keep people first and people know that you keep them first before anything else, that will change a lot for everybody. Not for you, but for everybody. Because they know that this person keeps me ahead of himself for everything, whatever is going to happen over here. And as long as they know that, they're fine. See, the thing is human beings are very simple creature in terms of trust.

Lot of time we want people to earn trust. I believe that once you hire somebody and make part of your team, they've already earned that trust. Just give it to them. Don't try to create >> Yeah, otherwise why are they there, right? >> Yeah, why are they there? Don't hire them else. If you don't trust somebody, don't hire them. Lot of people come and tell me, "AJ, set me up for success." Lot of people have told. And I told dude, or girl, when you we hired you you exactly knew what you are supposed to do. Yes. Then why don't you do that? If you do that, you will be successful. I don't have to set you up on some stage and announce your announcement. So, we never did that.

We never announce anybody saying, "Oh, this guy has joined in this No, no. Because they had organically grow. One more thing which we did at that that point, that every engineer had to go and sit at support for first 4 weeks. >> Every place should still do this. Like, this is good. >> Yeah. And once they are there, then they go and fix the bugs. And then they come out. Because what happens is once you do those kind of things, by the time they are ready to put you to you for you to put them on the some team, they actually already know most of the system and they already know who to talk to.

>> Cuz they've had the deal with it, right? Yeah. I think every place should do this. I think it's it's a no-brainer. >> So, own people, give them trust. One thing is that yeah, just give them trust. >> So, you moved from a CTO to being an operating partner in one of the largest VCs kind of in the region. Give us some What surprised you being in a role like that? Cuz it's very different than your builder role, right?

And what did you learn from all of that time? >> It good or bad, right? No engineering is no engineering. >> No, it's it's it's it's good and bad both. Okay, let's say let's not let's I'm trying to think more positively. Let's say it's good and not good both. Some things you miss, some things you don't, right? So, one thing which I really loved is that the exposure I had. So many startups, so many people are bubbling with ideas.

And we would meet so many people every day. Yeah. Wow, the world is amazing. What not so good about it was is that you are there on the field and hitting goals every day and passing to your colleagues, or passing it your friends, and making the team win. And then suddenly you will find yourself on the sidelines as a manager. >> And you're a spectator. >> Yeah. >> No, you're a manager. >> You're on the field, but >> You're not on the field. You're on the sidelines and you're trying to yell at them saying, "Please do this." So because you're because now my job was to go and talk to CPOs, CTOs, tell them what >> Does that mean you want to you see what they're doing and you're like want to help them.

>> You're doing what you >> what to do and sometimes they do and sometimes they don't. >> But they're not

necessarily inclined to listen. >> Yes, not always, right? So that was sometimes frustrating and sometimes very good. Some people just come and say and then you get feedback after six and also the feedback cycle is like really long. Sometimes you get feedback after six months. Sometimes there's somebody you're meeting at some meetup and say, "Hey AJ, six months back you told me this this amazing. Thank you so much. I did not do it earlier, but now it makes sense." So those kind of feedback loop are really stagnated, right? So that was not so good because you come from a world where you are making a code change or product strategy change or like product feature change and by evening you have the result. Because the product goes 2 and 1/2 million live people and some 30 million on in Jakarta or in Indonesia. So that was not so good. But good part which was there and that good part actually also sometimes makes you feel really sad is that, "Oh my god, they're doing so many things."

Why can't I do it? Should I quit and do? Should I quit and do?" So that also goes on. But what the most amazing thing was that you get to meet so many personalities. You learn so much about people. You learn that how we see life is like my investor colleagues, what they do. You also people have view about VCs, right? And I have seen both from >> You must have had one going in. >> I had a I had a view in going in.

>> I guess has changed being in it. >> has changed. There are So, they work hard. They work a lot cognitive There's a lot of cognitive load on the working. And most of time they're working with a very small team, one or two people. I So, there is a there is no big team concept. The office is big, the team is big, but every deal has one or two or three people, that's all, right? And then you are pulled in whenever your opinion is required, then you are not pulled in. But, what I have seen, what they produce, is fantastic. So, we a lot of time we think that VCs are very generalist, right? Because they can do biotech deal and database deal and as well as some cyber cyber spaceship. What? How come they can be so expert so experts on that?

They're not, but they're so curious that they want to be expert. They will go and read so much of stuff on a weekend, which we don't do generally. But, that's what their job is. That's why those people are different, right? So, I now have new-found respect for them in last 5 years, because I've seen them working closely. And then they have to maintain the people. They have to work with people in their good and bad phase both. If founder of a startup is not working, then it's still that VC is there to handhold him, explain him, either try to nudge him towards some direction. It's still there. So, it's sometimes it's frustrating for them as well, and sometimes it's very rewarding for them.

But, I think though that's what my That's what my new appreciation is for them. Earlier, I was like, why they are How do they make decisions? Now I understand, and there's a lot of There's a lot of art and science both goes into it. But, it's a lot of it is a data-driven, market-driven, and all this stuff. And lot of it it is not under their control as well. >> Yeah. Let's say that you're going to be returned to be a VC. Yeah?

What did you learn? >> If I have to return to VC, >> Yeah, to be a VC and now you're going to be the partner. Let's imagine. >> No. >> What what >> No, maybe. >> Yeah. >> That's what it's gone. >> Yeah, it's gone. Now, what I'm thinking for someone that listen and say, "Hey, I want to be like AJ, but I want to be the VC. How can I learn for you? Which type of mistakes or things that going to say, 'Hey, if it's post-mortem, I will do it better.'"

>> I think you have to be crazy focused and extremely analytical. That's what I think. That is a that quality. Relentless focus and extremely analytical. I think I'm relentless. >> I think I'm focused. But I'm not extremely analytical. Builders are usually not extremely analytical. Builders are Builders were on pay-as-you-go model. >> Did you and I'm not trying to dig and find dirt or anything. That's not where this question's coming from. It's more coming from You obviously hit a point where you're like, "I don't want to do this anymore or I'm out." Is it because you felt like it's not the type of thing you do and you want to get back to building? Is it like what prompted you to >> Many things. Like for example, let's say it was the same almost similar situation when I quit Gojek.

But this was This time it was something else. So, what I realized is that we we earn money and we spend money. We don't realize that we earn time with money as well. But we never get to spend time. And that's where the whole quest was for last 6 months or 12 months is that I have earned enough time. When I'm going to spend? And that was one thing. Second thing was I don't want to I didn't want to be with my kids. They're going through a very critical period in their life right now.

Young adult just hitting puberty, just going become a young adult again. I have two boys, right? One is 19, one is 40. And I have seen in Gojek when they are very young, very early, I was not with them. This time I want to be with them. You can argue that Ajay VC will allow you to do that. 100% like they supported me a lot. But then I what I realized is that I do not want to do this Oh, just wait for one more meeting.

I'll be there. I have earned enough time. So, let's do this. One thing One One was this trigger that staying with family >> Like I have a chance to >> I have a chance. >> spend more time with my family. >> Yes. And I can always go back to being operating partner or advisor or whatever that I am. That is was first thing is was that. Second thing which came to me was we are living in fantastic era of AI.

Amazing era. We can do whatever you want. We can work from anywhere. Right? So, if that comes to me after 6 8 months, then I'm free to do whatever I want. I can build whatever I want. I can be with people like >> So, a little bit of the builder itch. >> Little bit of builder itch. Yeah. But putting that on pause right now. And little bit of self-discovery itch right now. Just figuring out what what I want to do.

And people ask me Ajay, what are you building? And you will ask me as well. Ajay, what are you building? And I have very simple answer. I'm building my intuition right now. At this moment I'm really building my intuition. See, look what happens in our life. Lot of people don't think very seriously about job and relationships. When you get into job, you should think a lot. And fortunately or incidentally, I have always found my next job not while being on my current job.

Whenever I would quit my job, I did not have a next job. And I always found next job after my >> So, it comes in your dead time that you discover >> better because you are way more clear about what you want to do. And you are very more clear about what you do not want to do. See, in life So, I tell people that I'm cargo fit. >> But there is a risk. Some people will tell you, "Hey, quitting a job without knowing what you're going to do next, it's a huge risk.

How can I do that?" >> Because people do not want to be uncomfortable. People do not want to face this question in the morning that if I want to quit if I quit job today I'll forget fired. There's no guarantee, dude. Like everybody on 45 days notice or 30 days notice. Please understand this, all of us. So, what happens what makes your job permanent, right? That confidence. If you have that confidence today, then you can quit your job. Okay, so I always believed and I always still believe that you should be uncomfortable.

I was in ThoughtWorks for 10 years, but 10 years I did something different. Even in Gojek, every year was a farming formative year for me to learn something new. And the first year was like building a small team, second year building multiple teams, then suddenly if you become from CTO to group CTO because now you have 15 products in 9 months, right? And then you are the head of technology and all this stuff. So, if you're not uncomfortable with your job, then you are very complacent.

And if you're very complacent not content, you're complacent. That means it's business as usual where things happen or not. That means eventually people will you will disappear in the fixture. People may not realize that you exist. And somebody will notice one day and you're gone. That is what is happening with a lot of people right now who are at their 45s, 50, 52, 53 because they did not progress with the board and especially engineering and now 25-year-old, 30-year-old, 35-year-old can do the same job. What is the differentiation?

See, people don't pay you more for your skills. They pay you more for your experience and exposure. You go We all know that story of somebody came and somebody wanted to fix something and they hit with the hammer and he asked like \$1,000 fees. Why \$1,000? Like, where to hit the hammer. 999 to know where to hit. >> Yeah. >> Correct. >> To even know I needed a hammer. >> Yeah, exactly. So, I think that is what important. That's why I That's why I like people do say every time I quit.

Every time I quit, my friends always say, "You are nuts. Why are you quitting this job? Why are you quitting this job? There's no reason for you to quit your You Is anybody asking you to leave?" And I go, "No, nobody's asking me to leave." Why? Because I became too comfortable. And there's something >> it? >> Huh? There's too comfortable, and the people should realize that. >> You spoke about the hammer, and you like to punch the hammer on customer consumer retention, right? You said, "This is where it's important. Let's put the hammer over there over and acquisition, but then what happened is that you said that the viral spin it's have to be with the hacking road capability, but it's contradict.

>> Oh. >> How do you explain it? >> Yeah, that's There's context behind it. So, what happens is where every startup has this We hire the growth people. And my question to them is, there are two kinds of growth. One growth is you throw money and get more people. But their loyalty it sticks to you as long as you're cheapest in many cases. Or you're the best. Cheapest or the best. Cheapest, you buy with money. Best, you buy with features.

Right? So, we know that our customers churn out in 3 months, 4 months, 5 months. We know that they're acquiring a single customer, say \$5. We acquire 100,000 customers in a month, \$500,000 are wasted or gone. But those guys going to stay with us for next 3 months. But if we figure out why our customers churn out after 3

months and fix that, and then we have saved 50,000 customers don't churn out. We have saved 250k.

Which can be used to acquire new customers later on. So, that was the thought process. So, people take churn as a very usual thing. I don't take churn as a very >> Like they just accept and throw more money because I've churned rather than why are we churning? >> I'll give a simple example. Till 2020 I was there. Not only the boss but our team because of our team and all of the people around us we had the lowest attrition in industry. We had 3% attrition.

>> Of Gojek in engineering. >> And people who left us, they went to study outside or tried to go for the different kind of job. And we had very honest exit interviews with them and it was amazing. That is also That also came from churn. Every time somebody leaves us See, when somebody leaves you, they don't leave the company. They leave people, colleague, manager, environment >> or something. >> environment idiots, or whatever, right?

They leave that. Sorry for that. You can >> That's all right. So, the point is >> if you take the churn seriously then you can stop that. Same way with customers. If you If you can take churn seriously, you can use the same money in future to acquire more customers. >> Do you think the same model can be fit to B2B because you're talking about B2C? >> B2B is tough. Okay. So, B2B getting in is tough and getting out is tough. If B2B somebody churned out, you really did something absolutely wrong with them.

B2B customers don't churn out that fast. Unless you did something absolutely wrong. >> Let's see what's going on to Salesforce. >> Yeah, if for example Salesforce ERP or anything, it's very tough to get out of them. So, unless Salesforce did something or ERP did SAP did something or somebody else did something absolutely horrendous, then you will turn out like, "Okay, fine. I do not want that. I want that because this guy is like really pissing me off, right?" Something like that. So, B2B is a very different story. We should be very careful about what we are selling.

And there is a B2B software, right? When it solves the problem, either it solves complicated problem and also solves complex problem. Any software that solves a complex problem will stay. Any software that solves a complicated problem will get replaced at some point of time. Because what matters is a domain knowledge. Complicated domains are just a science problem or mathematical problem which you need to solve. It's complicated, one problem. Complex problems are actually juncture of 15 different problems coming together and humming as engine.

That's how I see it. >> So, let me ask you a more nuanced complicated question and I'm not asking you to incriminate yourself or where you worked or anything. So, just pull back from your historical references because the question only makes sense because you have all this history. Plus, you've seen the Indian ecosystem, I think, quite well given where you came from plus Peak. So, if you do that for a sec, I parachute you up looking down at the Southeast Asian ecosystem, what's your report card and learning from it? Because yeah, you have a few funds. Maybe even we could put Peak in the bucket that's done quite well. Have a lot of other funds that you know, the one the partners have never even seen a carry check, let alone cashed one, right? Lot of money put in.

What's your take? What could have been done better? What did you learn from it? Anything that you would say is "Hey, let me give you my take on this." Because I don't think I've ever heard it from you, right? >> Okay. Southeast Asia has come a long way. 10 years back, it was bustling. What is going amazing. Now that in in pockets it's going amazing, right? We had bunch of situations because of free money, no free money, interest rates, all that stuff impacted the funding sources and whatnot, right?

If you've seen funding winter versus funding summer or funding spring, you've seen last 5 years have told you told us every flavor of everything, right? Where we are right now, especially if you look at India, India has come a long way. First time in many many years, India has last two or three strong years in terms of IPOs. It's very fun. So, if you look at India, we had like last one or two years amazing years, right?

If you look at SEA, it is changing. Little bit. Like null period is going away. With advancement of AI and engineering cost going down, rapid iteration going much much affordable, it is actually much better system. If you ask me question, can we produce next Gojek out of here? I don't know the answer. But can we build from here for the world? That is definitely yes. >> Which wasn't really happening in my mind. Like even if we give kudos to the Grabs and the Gojeks and whichever ones we want to say, they're not building for the globe.

>> They were not yet. >> And they were building bigger things than other people, but very regionalized. The your last statement is one that I'm most interested in and cuz I get people that kind of rain on this right. Isn't it possible to be here with the IP and all the things that we have in this ecosystem and now AI and all that stuff to build for the globe? >> Yes, that's what is happening. If you look at If you look at recent lot of recent tech startups or SAS startups, they're like good example is Superbase.

Paul was here. And was here. >> Yeah. >> And now it's gone global. Right? While Paul and Ant moved and all stuff, which is I accept, but it happened from right here, right? >> And I want him on the show as well. >> You should call Paul. >> Okay. He's never around. He's in New Zealand. He's living on his island. >> But we we can You guys can call him. But I the only thing >> I would like I agree with you that they've done that.

>> Multiple >> But it really had nothing to do with the ecosystem, though. If I give you those two bookings >> even in the ecosystem. I'm talking about the I'm talking about the talent, and I'm talking about the possibility. >> Yeah. >> See, ecosystem is where you have to do build things only for ecosystem. I'm talking about like now is time for us to >> Like regardless of the ecosystem, build from here. >> Build from here. It's easy place to build from here, experiment from here, move to US, come to go to US, come back, go to US, come back, do whatever you want, and sell it.

It's easy. And there's no So, 15 years back or 20 years back when a SAS was just coming up, VCs were like very geographical, right? Now VCs can't be geographical. Even if they fund a company in India or Taiwan or something. Suppose they fund a amazing gaming company in Vietnam, and the game goes viral, the company goes viral. It is there in the whole >> Unless you're selling the security in Israel. They are only geographical. >> No, they are all over the place.

>> But the VCs, the Israeli VCs >> Because that is a tech ecosystem. Like same with Vietnam. Gaming ecosystem is in Vietnam, right? So, the I think the next SAS ecosystem can be in Singapore if people keep doing For example, we had a recent we had a we actually I had that was a news as well. Square X. >> Yeah. >> Yeah, was built here. Vivek built it here. >> Square X. >> Then he went there and he got acquired.

But he went there, but he built out here. So, my point is it's that zeal should change that looking at them locally versus looking at problems statement and solutions for global thing. And it's much possible the cost has gone down. You have \$200 cloud subscription, which is like relentless developer working for you all the time. >> Yeah. So, do you think we should expect more entrepreneurs from Singapore? >> People should do it. I'll tell you one more thing. People should do it. Second thing I want to talk about in this show is one more thing. If you know this whole labor market report from Anthropic and all this stuff. The most structured jobs going to get wiped easily.

Which is happening. We are seeing the job cuts like Jack has started with the whole Square and all this stuff, right? And that's going to just follow through all over the place. Every other company is doing it. So, what is we can do? What is our comp science grads can do? And there has to be something more. They have to go ex- get experience ex- get more exposure, do sales job, do whatever. But they are not entry-level programmers anymore.

Please understand this. You don't need entry-level programmers that much, right? Then it will and it should flourish more entrepreneurship. Instead of trying to get a job, why not create job for them? What way it is >> They're saying these things, and then the funny thing is I think it's the Ride Home show I listened to that the job market is suddenly seeing tons of growth in a lots of tech-related jobs, just not necessarily core engineer, right? That there that's You already see this because people can build bigger companies and things are happening, they actually need to hire people. It just may not be the engineer. It's like >> I'm referring to only structured like programming language is the most structured thing you can ever have. So that will get replaced.

Second thing accounting is again a structured debit credit debit credit all the time. Not the taxes but accounting. >> Yeah. >> Correct? That will get And that's how this whole ladder goes up all the way down to the whole way down, right? But the jobs like prompt will become the new programming language. How? And the to do that you need exposure, you need a product sense, you need to understand why we are doing certain things the way we are doing it, right? I have this When people want to meet me I have this shared calendar invite which says why what and how.

Earlier why was the founders, why we are doing something, what was product manager, how will it look and how I accept it. And how was engineering, right? That why what how has collapsed now. You engineering product design has collapsed now. So that means you need to have a little bit of everything. And also you need to have more intuition, more reasoning, more thought process driven thought process driven what do you say method to get things done than just being one siloed >> Yeah.

>> specialist. So yeah. >> You need more talent almost. >> Yeah. >> Almost done with time. >> Yeah. We should

Let's before we get to our closing stuff cuz we always do, tell us one thing you're building that you do want to share cuz I know you're building stuff and some stuff you're going to keep it to yourself but >> >> there's no way you got only one. We said only one. So just one thing. Tell us one thing you're building that you think people are going to hear about.

>> I I'm going to talk about it. I told you I'm building intuition. That is one thing I'm building, right? Because that is one thing very important. What I'm building, look, the reason I'm I do not want to say it because I'm just doing bunch of experiments to learn myself, right? For example, there can be so many small startup SaaS which has utility functions which people can actually use. So I'm building those. But putting them over here, I don't know whether it makes sense because may die may die tomorrow.

For example, I was experimenting with open claw, correct? >> We can put in the show notes that may die tomorrow and it's still we can preface the show notes. >> Yeah, may die tomorrow and will die tomorrow should not die tomorrow, whatever. Should not die tomorrow. >> >> So, basically I'm trying to see how I can help non-technical people with technology. So, there are two use cases. One is Hello AI. So, I'm building Hello AI along with the as a completely open source volunteer thing because it's very close to me on personal cause because I've seen that very closely. So, let me talk about little bit of Hello AI as well. So, Hello AI is a software which helps train people to deal with deal with kids, students, and everybody else if they're going through any anxiety, depression, figure out what is going on with them, how to deal with them. And I'll give you why. So, when you have pilot and he wants to fly the plane, you don't put him in the plane and send him to a sky.

You put him in a simulator, right? Because it's very risky because it can kill people. Same with the teachers and everybody else. So, this is going through pandemic right now. If you go through this, lot of work kids 15, 16, 17, 18, 19-year-old kids are going through this whole thing about anxiety, depression, ADHD, this that. Sometimes we just generally term them as oh, he's he's forgetful. He's not able to focus or she's not able to focus. And but there is a deep root cause. But teachers, therapists can't really Teachers especially can't really deal with it. Therapists can deal with it, but therapists don't have enough training to deal with them. So, what we are creating is a software which is completely open source which institutes can deploy at their end to create training for individuals. They can practice over there and identify this cause. And it's very close to me because I've seen it first row what it goes and then I have so many friends whose kids are going through this right now. So, I think that is good. That is one thing I'm building. And that is good. Second thing I'm building is and that is my builder's age.

Trying to see can people how can people use Open Cloud how can somebody actually non-technical use Open Cloud because configuring it is very crazy and may die tomorrow by the way. So I'm building a SAS for that. I'm helping people share their photographs especially in India. They don't share photographs. They share photographs on WhatsApp but they don't create album because they never subscribe to any of these things. How do you do that, right? Or many other things. So basically one is purpose which is Hello Laiola or other other institutes I'm advising. That is purpose-driven. Second is just builder's age.

>> Being old as heck. >> Yeah. Just just that. >> Yeah. >> It's fun. The thing is again I told you know that

perfecting the perfection. >> >> No, no, I think you're having fun, right? I think it's good to see you having fun. So >> Yeah. Yeah. >> Go. >> Okay. >> >> What is the one thing which is a life tenant that you're going to tell to a 20-year-old that might going to adapt? >> Okay. There are two things I lived by.

One is earn money. Earn people not money. I always tell people earn people not money. And second thing I always lived by is ideate, implement, iterate. If you are trying to do something, first yeah, a lot of people going to do job and all this stuff. But always ideate ideate, implement, iterate on many many things. Keep doing the experiments in your life. Because this time they have a lot of time. Now if I tell myself my 20-year-old myself saying you wasted a lot of time.

And they should realize this but they are not. They are not. So I would say ideate, implement, iterate as much as you can and you have a lot of time. Remember that you have a lot of time. >> Yeah, and use it. Use it. >> So I've known you for a while and I know that you went on a journey to help your own health and kind of get to where you wanted to be. So, tell us a little bit about what you're doing for mental, physical that you finally have time to focus on it.

>> Two things. One is a diet. So, I'm fixing my diet. Trying to eat very sensibly, but I'm trying to eat everything. So, I'm not trying to say I'll not have sugar, I'll not have carbon. No, I'm doing everything. So, I'm going back to the 20-year-old diet. What is 20 20-years-old? People used to have everything and they were healthy. Something what has happened in last 10-15 years, the abundance has come to our life. So, we just eat the things which are abundantly available to us and we can purchase it and afford it. So, we have forgotten the balance of diet.

Correct? First thing is that. So, I'm going back to that. Second thing is I have learned over last 7-8 years, especially after Gojek, is that taking care of yourself is super important. So, exercise is good, but mental health is much, much better. So, how do you Okay, there's a I'm going to take a little bit of detour. >> Go ahead. >> So, there's a very good tagline I have learned recently. It's called here we are. Here we are. So, a lot of time your kids don't listen to you, somebody else's kids don't listen to them and all this. So, I was talking to one of my friends, his name is Andy.

He's in Chicago, near Chicago, Detroit. And he visited us back then. And then he sent me a picture of his son one day and his son has a long hair. So, we were talking about like his son should cut his hair and all this stuff. He had a hair cut. And so, Andy responded very nicely saying, "Look, I told him, but he doesn't listen. And here we are." >> >> And that is so nice. If we just say here we are for many things in life, we'll not stress about it.

So, if somebody like you don't listen to me, okay? Nice. I told Mike he should do something. He did not listen to me. Here we are. And as soon as we here we are like, okay, we can't do much about it. >> That's interesting. >> And once you can't do much about it, once you accept that, the biggest problem we are going through, all of us going through, we think we can really fix things. While 99% of things which we do today are not in our control. I'll give you a simple example. A lot of men or women in 40 getting up in the morning on Saturday morning and going for groceries and coming back. Just because they got married and have kids.

So they have what they did 20 years back. That's why I think they're getting up today. While somebody who's bachelor doesn't have to get up. So we do a lot of things today because what we have done in past. So that's not in our control. It's gone. This country accept that accept as it is. So I'm going through that journey. Being patient. Accepting life as it comes. And not fret about it. Because I realize one thing in Go-Jek, whatever can be done at 9:00 p.m. in the night can also be done at 6:00 a.m. in the morning.

>> Yeah. Not much difference. >> Not much difference. Don't waste your night. And I tell everybody this. >> It's super powerful and it's and I've been trying to learn this as I have a teenager and actually two teenagers and >> you patience. >> Yeah, that is just I've said my peace. I feel very strongly about how I feel about a particular situation, but I can't do anything about it. Like I can't fix it. I can't make them do something and I think it's a great bookend to just say here we are. Like I've said my peace.

>> can do it in any age. I do it with my four years old. I do it with my eight and my 11. >> You can do it with anybody. >> Everyone. Anyone. >> Point is that you need to understand that tipping point where we say, here we are. >> Yeah. >> A lot of time you just don't give up. >> It's a great way of framing it though. I like it. That's a nice trick. I'll be using that one. Yeah.

>> Okay. Do you read some things or watching some good things that you want to share with us? Maybe a good book or whatever. >> Oh, good book. I think in the age of AI everybody should read this book called Philosophy of Software Design by >> I'll find it. Don't worry. Just What have you got there? >> John Ousterhout, something like that. >> It's old, right? >> old. Read that book. That is well that will help people a lot. And then every >> Because the principles still apply.

>> Principle has to apply. >> Yeah. Whether there's something generating code for you or not. >> Yeah. >> Yeah. Principle has to apply. If you don't have exposure to what is what's going on, after 12 days of white coding you're totally clueless what the hell happened. >> It's still working. >> Yeah. People have lost their environments, bro. Haven't you heard those news? Like entire environment like >> Now my big problem now is I'm using like Codex 54 like super high and like it's always compacting the memory thing and I forget to You're supposed to write things out for it because it wakes up and it drifts. So then I had to realize, oh god, I got to tell it to like >> Like send them out of space again. You are this This is our environment. This is where this this this this is You have to do all this.

>> it drifted again. Yeah. >> So I think John Ousterhout, I think so. Find it up like >> I'll find it. I'll get it in the show notes. >> So that's that one book I always read out of that book. Over a period of time, I have learned Gita is a very good book. Leave the spirituality apart. Leave the religion. Lot of scientists have read that book. And I'm reading it nowadays and lot of good things out of it and just take that about how to deal with people, what is there, what is not there. It's So we have a very big historical mythical scripture called Mahabharata. One of the chapter is actually Gita where Krishna goes and tells Arjun about how to behave, what to do, why Arjun refused to kill his brothers, cousins, and all stuff. And Krishna is telling him like, this is your duty and the whole dialogue is between both of them about about what your dharma is, what your karma is, and what you should do, what you shouldn't do. That is a very good book.

I read it long back, but I'm reading it again. Nothing to do with religion. Just read it. It's good. Third book which I always liked, and in terms of like just just as what do you say? Just as science fiction or whatever. Like any of Asimov's book. >> Yeah. >> Yeah. And you people have seen their movies and the movies like I, Robot and Bicentennial Man and all stuff. If you want to read those books, please read those books. Those are classics.

>> Yeah. Yeah, self-help and all the stuff, I would suggest for that talk to your friends. >> So, here we are is probably the biggest self-help >> we are. Yeah. >> Here we are. >> It's a good one. >> Tell us in right there. >> It is it is pretty very amazing stuff, right? >> >> And the Ray Dalio Principles. >> Yeah. >> Very good book. Very good book. I love that book. >> He's had some good posts the last couple days with all these Middle Eastern stuff. It's been quite interesting.

>> It has been quite interesting and I'm like so disappointed in this humanity. >> Yeah, it's sad. >> Like why? >> Yeah. Okay, we got to leave on a good note. So, we're going to keep an eye on these things you're hacking on and if one of them is still alive by the time we post, we'll get it in the show notes. Other than that, we're going to keep an eye on you. We know you're going to be doing other things and thanks for coming on the show finally.

>> Thank you. Thank you so much. >> I really appreciate you giving me time. It's like amazing to meet you. We're here and I like I really enjoyed this conversation. >> Yeah. Awesome. Thank you. >>