

Microsoft CPO: If you aren't prototyping with AI you're doing it wrong | Aparna Chennapragada

61 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=HBBFXAWCUU0](https://www.youtube.com/watch?v=HBBFXAWCUU0)
<https://www.youtube.com/watch?v=HbbfXAWcuUo>

SUMMARY

Aparna Shanapraada, Chief Product Officer at Microsoft, discusses the evolving landscape of product development, emphasizing the importance of prototyping and the integration of AI in the product lifecycle. She highlights the shift from traditional user interfaces to natural language interfaces (NLX) and the role of product managers in navigating this new terrain.

- *Prototyping is essential in modern product development; teams should build to visualize ideas.*
- *NLX (Natural Language Interface) is becoming the new standard for user experience, emphasizing conversational design.*
- *The role of product managers is evolving but remains crucial for taste-making and guiding product direction amidst an influx of ideas.*
- *Companies must balance rapid prototyping with careful scaling, as the time to demo is shorter but full deployment takes longer.*
- *Collaboration between humans and AI agents is key for future productivity, with potential for new workflows and efficiencies.*
- *The importance of understanding underlying technology shifts, consumer behavior changes, and business model innovations when launching new products.*
- *Emphasis on resilience and adaptability in product development, especially when initial ideas do not pan out as expected.*
- *Coding remains relevant; higher abstraction layers will emerge, but foundational knowledge is still valuable.*

I have a cheesy Chrome extension. Literally, whenever I open a new tab, it just says, "How can you use AI to do what you're going to do right now?" How do you see the future of product development being different? If you're not prototyping and building to see what you want to build, I think you're doing it wrong. It becomes even more important to have the editorial and taste making at the heart of it because otherwise you just have a Frankenstein product.

There's this acronym that you taught me, NLX. What is that? Natural language interface. NLX is the new UX. Often I hear a product builder say, "Oh yeah, with AI like the model eats the products." That doesn't mean it's not designed. You and I are having a conversation. It's a podcast. I'll have another conversation at Microsoft and that's a meeting. Conversations also have grammars. They have structures. They have UI elements. They're invisible. What are the new principles, new constructs in natural language as a interface? I just saw that cursor

hit 300 million ARR in 2 years.

Interestingly, you guys were very well positioned to do really well in this AI coding tool space. You guys said co-pilot, the first tool in the world at this stuff. So, ahead of everyone. What happened? I would say today my guest is Aparna Shanapraada. Apart is chief product officer at Microsoft where she oversees AI product strategy for their productivity tools and their work on agents. Previously, she was chief product officer at Robin Hood, vice president at Google, where she worked on Google Lens, search, shopping, augmented reality, AI assistant, and a lot more. She was also a longtime engineering leader at Okami, and on the board of eBay and Capital 1.

In our conversation, we chat about how working in B2B is like being JeanClaude Vanam doing the splits across two moving trucks. how she's operationalizing her team living in the future so that they're building towards where things are going, why people still need to learn to code, why the PM roll isn't going anywhere, why NLX is the new UX, and so much more. If you enjoy this podcast, don't forget to subscribe and follow it in your favorite podcasting app or YouTube. Also, if you become an annual subscriber of my newsletter, you get a year free of a bunch of products, including Linear, Superhuman, Notion, Perplexity, and Granola. Check it out at lennisnewsletter.com and click bundle.

With that I bring you Aparta Shenapraada. This episode is brought to you by EPO. EPO is a next generation AB testing and feature management platform built by alums of Airbnb and Snowflake for modern growth teams. Companies like Twitch, Miro, ClickUp, and DraftKings rely on EPO to power their experiments. Experimentation is increasingly essential for driving growth and for understanding the performance of new features. And EPO helps you increase experimentation velocity while unlocking rigorous deep analysis in a way that no other commercial tool does. When I was at Airbnb, one of the things that I loved most was our experimentation platform where I could set up experiments easily, troubleshoot issues, and analyze performance all on my own.

EPO does all that and more with advanced statistical methods that can help you shave weeks off experiment time and accessible UI for diving deeper into performance and out-of-the-box reporting that helps you avoid annoying prolonged analytic cycles. EPO also makes it easy for you to share experiment insights with your team, sparking new ideas for the AB testing flywheel. EPO powers experimentation across every use case, including product growth, machine learning, monetization, and email marketing. Check out EPO at getepo.com/lenny and 10x your experiment velocity. That's get [epo.com/lenny](https://getepo.com/lenny). This episode is brought to you by Pragmatic Institute, the trusted leader in product expertise.

Pragmatic Institute helps product professionals turn ideas into impact through proven courses, workshops, and certifications designed for real world success. For over 30 years, they've trained more than 250,000 product leaders at companies like Google, Microsoft, and Salesforce, equipping them with practical strategies to build and scale marketinng products. Pragmatic's full-time instructors each bring over 25 years of hands-on leadership experience, teaching strategies proven to deliver realworld results. And it's not just about what you learn, it's also about who you learn it with. Completing a course connects you to an active community of over 40,000 product professionals. You'll engage in meaningful conversations, collaborate with peers and mentors,

and gain direct instructor access to refine your strategies, and stay ahead of trends. Get 20% off with code lenny 20 at pragmaticinstitute.com/lenny. Aperia, thank you so much for being here and welcome to the podcast. Thank you, Lenny. Thanks for having me. when I asked a lot of people that work with you what I should ask you about and what's what I should know about you.

Something that came up again and again is something that I think most people don't know about you, which is that you're you're big into standup comedy and you take it semi-seriously? just how serious are you about this? How how much of your life is this? And most importantly, how does this help you build better products? It's hard to say I'm serious about like a funny business, but I I do I do watch and do standup comedy. I do open mics. I've done a few shows. Wow. I have one set brewing that is around u AI unsurprisingly AI and tech and Silicon Valley. You know, it's really interesting for me. This was an accidental discovery. Like I'd always been an SNL fan and like just comedy fan, but I went to an open mic because, you know, my son sings and he went to the open mic for singing and he's like, "Mom, you should go do this." And I was like, "Oh, let me go give it a and I found that I enjoyed it and was good at it. To your question though about building better products, I'd say both have PMF. I mean product market fit, punchline market fit, punch line market.

but I actually there are a couple of things that I I do find really powerful and useful because you know in open mics or even when you're testing these things it's a very tight cycle of iteration and you get live like open mics are the real live experiments right you you put something out there you get very clear micro feedback from users and then you get tough feedback sometimes and I think as product builders that's actually one of the great skills to have which is yeah you you sometimes launch stuff that you know have a fantastic vision but the first version is not quite there right and Reed Hoffman says this hey if you don't launch the first version and are not embarrassed you're doing it too slow just that gap and closing that it's good resilience yeah I never saw these coral areas between these two things I didn't realize you actually did like shows and you're working on a set I wasn't going to ask you for a joke but if you're working on on a whole thing about AI is there something that you can share from that set one joke I'd maybe share is people think about these AI chat products as women because you know you don't know what's going on.

It's a black box and you don't know what it's what what they're thinking. There's like an entire set around that. But obviously on the flip side too that you know they're probably more like men in the sense that they hallucinate a lot. They're they kind of are not yet reliable. I'm afraid to laugh at this a little bit. This is great. Okay. and they even when they don't know the answer, they make up stuff. They're very confident.

This is good. Where are we going to be seeing the show, by the way? okay. This is great. Okay. let's get serious again. So, you worked at most of your career at a lot of consumer internet companies. You worked at Google, Robin Hood. You're on the board of eBay, on the board of Capital 1. Now, you're at Microsoft. I'm curious just what is most different about working at a company like Microsoft and building product at a company like Microsoft? I think intellectually I knew that hey enterprise particularly the the area that I look at most at Microsoft is focused on enterprise and productivity and transforming companies through AI.

And to me I I think two things really strike as very different. One, in fact, I just posted about this the other day

saying in consumer, you're kind of like, oh, you have a playbook for make the product work or make the feature work and make it delightful. But I think in the enterprise, you almost have every time you you think you have one use case, you really do, which is how do you make sure that the feature works well and there's governance of the feature, right? If you think about like even something as simple as sharing a link to a document, you want it to be easy, frictionless, but at the same time, you want that to be secure and kind of safe and being able to have auditability and all of those things. And often I find that when you go from consumer to consumer to enterprise, you fall into a trap of either disregarding that, right, and say, "Oh, you know, we'll just focus on one side of the house," or kind of overly crippling the user experience, right? and kind of you know leaning on the other side. So I think the there's an art and science and nuance and playbook there too. So that's one big learning for me. The other learning and especially in the AI era for me has been about this you know I think there's a famous trailer from the 2000s on Vanam on these like two trailer two buses doing Yeah. doing the splits. Exactly.

And I feel like a lot of the the companies including the tech companies but certainly the enterprises that I talk to are in these two modes where one hand this is the most compressed tech cycle that we've ever experienced right it's all in the order of weeks and months versus years and decades if you think about like mobile and cloud and internet and there's just like so much happening the intelligence overhang on the other hand there's also like humans and habits that productivity habits change it's hard to change and change management through the company is also hard right you don't want to kind of be rash on that so it's like you know the future is unevenly distributed but even within the companies on the second bucket of this other this the the bus that Vanam's riding on of governance and I don't know adoption and changing behavior and stuff is there something you've learned about how to get past that help help that along more the thing not to do is hold back folks who are early adopters, right? I think that's the that's the other one learning. In fact, I think that's one of the reasons why recently we you know I've been working with folks to say can we have can we have both which is the longerterm change management being able to do it in a trust trusted way at the same time do this program we're calling frontier program and roll out cutting edge experimental features. We learn we just built this world's first agent for deep research agent made for work right post train for work and of course it has you know all sorts of edges rough edges but if there are early adopters in an enterprise or outside how can we kind of put that in the hands of those folks without kind of insisting that all of the all of the company be completely different developing different muscles this program Frontier you're you're talking about I wanted to spend a little time on it so what is the idea the idea there's like people are working in this futuristic environment. How how does that actually work? Yeah, I think the idea is exactly this which is like I want to kind of institutionalize and operationalize my personal model of like living one year in the future and say what does this imagine a company or a setup like Frontier in Consulting Group or Frontier Inc. Right. And if you did lived in that environment where you had all the AI tools and really advanced deep research intelligence on tab, what are the kinds of questions you'd be asking? What what's the kind of work you'd be doing?

How would you change how you're going about your workday? so that's the premise and you'd say, hey, how does it change an individual? But also down the lane, we want to think about what does a frontier team look like? We talk a lot about Frontier Labs and models. I think models layer is amazing and obviously like you know that's what empowers all these product building to happen but I want to push us to think about what does a frontier product look like and more importantly how does a frontier way of working right like what does a team with three people and tons of like compute and AI tools look like so how exactly does this work there's like a

team within Microsoft that's like your job is to use all of our latest tools and build product using that. Is that Yeah, that is that is the setup. We're just a few weeks into that setup. But meanwhile, what we've done is like we've actually set up a like a comp external like a fake company and said, "Hey, if you are somebody who wants to come play with some of the cutting edge science projects and deep research agents and you know agents at work, come party here." H wow. Okay. And it's only a few weeks in. Okay. So TBD how it all goes.

Yeah. Yeah. And again like these are micro that's the the meta point here right also is that you know in the traditional way we've kind of always thought about across the companies across industries really thinking about rollouts in these macro ways right you build something and you kind of like roll it out you have a general availability for and then you take the time and that's really important too because again like we're talking about pharma companies legal companies relying on this so we do want to have that but at the same time given the compressed cycles of AI How do we start to have people experience what's what's the one year in the future? Let's follow this thread in a few different directions.

There's like how product change development changes. There's how engineering changes. There's also just agents. I know you're spending a lot of time in agents. Feels like you're not an AI company these days if you're not working on agents or building an agent. Lenny, we're doing this wrong. We didn't you didn't use the word agents so like so so far into the conversation. I I try hard to push it out as far as I can. It's like it's like every conversation in San Francisco just like how long until I start talking about AI. Yeah.

It's like three minutes average, I bet. Oh man. Okay. So, so with agents, I know that you're leading a lot of this work at Microsoft and a lot of people are wondering what the hell what the hell does this mean? What is going to change? Give us just a glimpse into how you see the world being different in a world of agents being around more. Look, there's a short term and there's a long term, right? There's a lot of, you know, hyperventilated excited talk about kind of the the eventual future and all of that. I take a much more practical product building lens on this, right?

And I think about these at the end of the day, they're tools, right? Yes. Underneath it, there's stochastic models versus very deterministic programming tools. You can tell I'm a computer scientist. Like the way that that worldview definitely shapes how I think about this. To me the short term is there's an evolution like we had apps right and now I think we are firmly in the assistance era where there's like human driving the you know that's what we think of as co-pilot right like I think the human driving kind of the in the driver's seat but having a lot of assistance from AI so I think of this as then you you you look at the dimension of almost like autonomy and delegation and intelligence as the intelligence for example when deep reasoning unlock happened. Of course, then you could say you can delegate more right to to the agent. So I think to me I think there's one dimension where you say hey agents are somewhat independent software processes right that can kind of like run tasks and you're not just thinking about hold handholding and fine motor stuff. You're saying hey here's my goal go make this happen. like I'll give you an example right so we're working on this researcher agent for work and last night I said hey you know I'm really I I have an important meeting coming up with the leadership team I really want to present these frameworks here and this is the road map here go back and look at all the people that are in the meeting what are their views on this topic and kind of come up with how do how I should kind be

thinking about like you know the right persuasion pitch here right and what's magical about this is not just that it's saving time. Typically, we think about the Sofar AI as summarizing a document or saving time, right? This is like firing synapses that I didn't I didn't quite have and actually giving me new insights and giving me dare I say superpowers, right? So, that's a natural evolution of AI, I would say. So, when I think about agents, I think about three things. one is is an increasing level of autonomy and kind of independence that you can delegate higher and higher order tasks. Second thing I think of it is complexity, right? So it's not just a oneshot, hey create this image or do this thing or summarize a document. It's you know build me this prototype that expresses my idea of an augmented reality app, right? it's a complex task. And then the third thing I would say is asynchronous. It works when you're not working, right? I think that's the other big thing about these things that you're not have you don't have to sit in front of it. This is answers the question of what is an agent? essentially these three bullet points. So it's order the three again when I think about agents I think about these three things right so one it's autonomy like being and it's a it's a spectrum it's not a zero one it's how do I actually delegate things that it can do second I think of as complexity right it's not a one shot hey summarize this document generate this image but it's you know build me this prototype or help me knock this meeting out of the park right and then the third one I think of is it's a much more natural interaction that doesn't just mean chat, but it may be actually jumping on a meeting with the agent and being able to like talk through all of it or point it to things that I wanted done differently. So, I think all three things, the autonomy, the complexity, and the natural interaction are at least product principles that will shape really good ones, good agents. That is really helpful. Along this line of agents, there's this acronym that you taught me as we were chatting ahead of this podcast, NLX. What is that and how does that relate to agents and why are people not thinking about this enough? Oh, that's one of my Roman empires these days. The the natural language interface NLX is the new UX, right?

so I think here's the here's the here's the deal to me. I think traditionally we've thought very consciously about gues because the graphical interfaces are not something natural and so they have had to be explicitly designed but they're rigid interfaces, right? what we have with conversational interface and natural language is it's a it's a much more elastic right that doesn't mean it's not designed so people have often I hear product builders say oh yeah with AI like the model leads to the product so it's just you chat with it you and I are having a conversation it's a podcast I'll have another conversation at Microsoft and that's a meeting so conversations also have grammars they have structures they have UI elements they're invisible And so one of the things that I see and I'm really excited about is what are the new principles, new constructs in natural language as an interface.

I'll give you a few examples, right? And actually like lot of startups as well as big companies are really experimenting with this stuff. One is if you think about it, prompt itself is a is a new construct and that's a new way that's a new UI element just like a drop-down was or a menu was. But others that are emerging especially for agents I think are plans. So when you give a high high level goal, what we are seeing is that when the agent comes back with a plan, preferably an editable plan, that's a new construct, right? The other one that's that I think about a lot is showing the work, right? Progress. You see this with different products, right? You see with the co-pilot, you see with JBT, DeepC, this idea of thinking aloud and it's kind of showing the work. but how much do you do it?

If it's too verbose, it feels like I'm running some crown job and scripts. but if it's too tur, then I don't know if it's going in the right path and I don't have the confidence yet. So there are all these new elements. So if you're a

product builder, this is a fun new space to be digging in for product design. This is really interesting because I think people chat with all these chat bots and it just feels like this is just the way it is but you actually are designing every element of the interaction like how much to share but how much you're thinking here's the my plan what do you think so I think I think this would surprise a lot of people just realizing there's so much that goes into just designing even these what seemingly are simple conversations yeah another good example is follow-ups right you could say look you have a you you ask me a question and then I I could ask a follow-up u set of things and that's explicitly should be designed for success right so for example if I said hey create an image and it created a black and white you know I don't know like a a clip art version of something what are the next obvious follow-ups that it should be suggesting proactively now too much and you're kind of annoying me right like but too little and in in some sense you've lost an opportunity to direct me or guide me into a happy path here. This resonates a lot with when we had Kevin Wheel on the podcast, he talked about this question of just how much to show about what you're saying and you know and it's interesting that Deep Seek went the extreme of just showing everything and people liked it too. I think that was interesting. Yeah. And I think it's a point in time thing tool, Lenny, because in some sense right now, these things are such black boxes. They're almost like peeking under the hood for anything, even if it's verbose. Feels like, oh, I know what's happening.

Especially because the compute inference time, it's taking long to think. so it just feels like if you just went silent, I'd be very uncomfortable. I think exactly. so I do feel like there's that point in time but over time I also feel like this is an area ripe for personalization for example right like again in human like my API would be very different from somebody my interface is probably different from others and I might just want the direct hey give me the TLDDR versus the oh so I went here and then I went there and like following the thread a little bit we're talking about just how the future is going to be different there's like designing for these chat experiences there's agents. kind of zooming out to just product development in general.

It feels like you're at the forefront of a lot of the tools that are going to change the way we build products and also your teams are working with a lot of these tools that no one else has access to. So, let me just ask, how do you see the future of product development being different from today most and what do you think product builders should be preparing for doing to kind of to succeed in that future?

I will I'll start with one stark statement that I say internally and externally and I I'm trying to live it is that in this day and age if you're not prototyping and building to see what you want to build I think you're doing it wrong. I call it the prompt sets or the new PRDS, right? Like I really insist on folks saying if you're building new projects, new features of course come with prototypes and prompt sets. And I think the the the the notion is not to say hey now like everybody is just you know like a biggest version of a like a software engineer, right? it is to say you know you have the fastest path to kind of seeing and experiencing what's in your mind to to be able to communicate right it's a much more high bandwidth way of communication I think about that as a really a loop accelerator in terms of product building that's number one when in doubt as someone put it demos before memos right I think like that's that's really number one I would say number two this one is a little bit tricky I'd say is that what I'm seeing is that the time to first demo the is much shorter, right?

But the time to like a full deployment is is going to take longer. So I I think that there's going to be an uneven

cadence. So typically I think there was much more of a hey you build this thing, you take a few weeks and then you kind of iterate and so on. now but that inner loop of like prototyping and iterating and getting even user research through AI conversations all of that gets shortened but I think the bar for scale therefore becomes much higher right in some sense if you look at it like there's going to be a supply of ideas right like a massive increase in supply of ideas in prototypes right and so which is great it raises the floor but it raises the ceiling as well right in some sense like how do you break out in these times that you have to you have to kind of make sure that this is this is something that rises above the noise. So I would say that it's simultaneously thinking about not chasing after every idea like right I think is the second one. I'd say the third thing is you know there's a lot of conversation around full stack builders right what does the team of the future look like bu product building team what I think about is I think that is inevitable in terms of like there will be a few folks that are especially at the prototyping early idea discovery stage that the lines are blurred right you have there'll be a few taste makers at the same same time I think you can still have a lot of people experimenting it becomes even more important to have the territorial and taste making kind of u one or few at the heart of it because otherwise you just have Frankenstein product right that that definitely doesn't change I have one other additional bonus thing which is a lot of folks think about oh you know don't bother studying computer science or you know the coding is dead and I I just fundamentally disagree if anything I think you know, we've always had lay higher and higher layers of abstraction in programming, you know, like we don't program in assembly anymore. like most of us don't even program in C and like and then you're kind of, you know, higher and higher layers of abstraction.

So to me, there will be ways that you will tell the computer what to do. right, it'll just be at a much higher level of abstraction, which is great. It democratizes you. there'll be an order of magnitude more software operators like instead of SWES maybe we'll have souls right but that doesn't mean you don't understand computer science and it's a way of thinking and it's a mental model so I I strongly disagree with the whole like coding is dead that's awesome I love that and so is a software operator is that what that stands for yeah I just made it up but yes okay cool this idea of prototyping as being kind of core to building these days. Is there anything you do within Microsoft to operationalize that and make that just like a thing everyone has to do? Is it just like culturally do it or is it like you must show me a prototype before you show me it? You know, I think it's again like the future is here unevenly distributed even in Microsoft I would say. But there is certainly a strong cultural momentum and shift and desire to say hey let's let's actually look at live demos live prototypes and to even like communicate the ideas right and and to me I mean it's not always possible because obviously there are like things that are deeply like if you're trying to change something in like the bowels of Excel you probably don't there's even enough depth in the in the product that you know what you need to do and you don't need to prototype that. But if you're especially thinking about new things, new products, new features, absolutely.

Okay, let's talk about product management. There's this fear that emerged as soon as all these AI coding tools came out of just like PMs are dead. We don't need PMs. We could just build things ourselves. What what are these people hanging around for? And what I found is it's actually the opposite that now that coding is easy. Now the question is more and more what should we be building why should we be building it is this right is this the right solution then getting adoption for it which is what PMs are really good at and so I feel like it's the opposite like PM are the most important role and there you know it'll change too but but let me get your take just what do you think the future of product management looks like you think it's dead do you think it's going to thrive do you think it's going to change yes meaning meaning look I mean if you're a TPS report mostly process

person and like a lot of companies do get confused about product management and process and project management. I think then you do have a question of like hey what is the value add here right especially if like AI can read and write like 50,000 meeting notes and you know track things and send emails and so on. but I think what what I do think on the flip side is the taste making and kind of the edit editing function becomes really really important right in a world where the supply of ideas supply of prototypes becomes even more like an order of magnitude higher you'd have to think about like what is the editing function here so that does mean that the bar is higher for for you for product folks but I think there's a there's an interesting side effect I am observing in you know startups that I'm advising companies and even within the companies that there's there used to be more gatekeeping I would say in terms of like oh this is you know we should ask the product leader what they think and again like there is a role for that editing function but you have to earn it now you just don't get it because of this title but there's also just like unlock of latent really good ideas from smart engineers smart user researchers smart designers who can now who now have like this expert in their pocket, right? To kind of round out all the other things that they're not they're not typically skilled at to bring forth their ideas. And that's f amazing, I think. And I think that expert it's interesting I'm working with an engineer and some stuff and he uses chat GPT to even communicate to me in a more effective ways like turn this pitch into something that will convince Lenny this is a good idea. By the way, that is actually one of my common use cases which is WWXD. I call it what would X do like I use to say hey what would Satya think about like this particular set of conversations or ideas that we're pitching and so on. This is the power of like I think deep praising plus relevant context, right? This engineer you're talking about has that context about you and so it's kind of very interesting. If only everyone was as famous as Satia and had so much information out there. But I guess you can import all their emails or whatever tools exist to just like understand from the conversations you've had with that person. Yeah. And I think this is this goes back to actually what you were saying too which is I think this idea of what is the there's like a coil spring.

There's an intelligence overhang that I I just see across the board. And I think the part of product development has to almost rewire ourselves to I think Toby from Shopify calls it the reflexive AI usage. and that's not as easy. And I've been thinking about why. Like I basically I mean I have a cheesy Chrome extension. Literally whenever I open a new tab it just says how can you use AI to do what you're going to do right now. Just like it's very cheesy but it kind of helps to pause and think oh what what what am I trying to do here? But the reason I find it hard and when I talk to even like people who are living and breathing in this space they find it hard is that you know the updating of the priors is really hard like the models couldn't do some things one year ago like I mean image generation was full of spellings or like reasoning you just couldn't like you know have deeper and smarter answers it couldn't do data analysis so like my impression of it from change trying it a few months ago that prior needs to be updated and it's hard to do that right and you have to kind of do something almost counterintuitive and against the grain to say no no no like ignore what you learned about like what this can or cannot do like the baby just grew up to be a 15year-old in a month. I think that last point is so important that we've tried these tools over the years and it many like so far it hasn't been amazing and then all of a sudden it is and you kind of don't know that and you've given up almost and and things change. I think that's actually if you're a product builder listening to it, that's a really interesting arbitrage thing for you.

Like if you can kind of cut against the grain and say, "No, I won't have that scar tissue around like you know this didn't work a few months ago and keep setting high expectations and like demand more of the AI today." I think

you can you can unlock more. There's a lot of alpha in in doing that. That's right. Today's episode is brought to you by KOD. I personally use Koda every single day to manage my podcast and also to manage my community. It's where I put the questions that I plan to ask every guest that's coming on the podcast. It's where I put my community resources. It's how I manage my workflows. Here's how KOD can help you. Imagine starting a project at work and your vision is clear. You know exactly who's doing what and where to find the data that you need to do your part. In fact, you don't have to waste time searching for anything because everything your team needs from project trackers and OKRs to documents and spreadsheets lives in one tab, all in KOD. With Kota's collaborative all-in-one workspace, you get the flexibility of docs, the structure of spreadsheets, the power of applications, and the intelligence of AI, all in one easy to organize tab. Like I mentioned earlier, I use Kota every single day.

And more than 50,000 teams trust Kota to keep them more aligned and focused. If you're a startup team looking to increase alignment and agility, Kota can help you move from planning to execution in record time. To try it for yourself, go to kod.io/lenny today and get 6 months free of the team plan for startups. That's coda.io/lenny to get started for free and get six months of the team plan. kod.io/lenny. I'm going to come back to this cheesy plugin. Say more about this. So, this is a plugin that just lets you put a custom message on every new tab and it just you have it say how can you use AI to do this? Yeah, it's as as cheesy as that.

And it's interesting because it works. in the last few weeks alone, I've been doing this like experiment to say, hey, how much more AI pill can I get? like both at work and in a personal life to say, you know, when I'm trying to do anything manual, like should I be demanding the AI to do this? That's so cool. Do you know the name of this Chrome extension by any chance? Otherwise, no, I built it. You built the Chrome extension? That's so cool. okay.

did you use AI to build it? Of course. Wow. Which tool did you use to do that? Some kind of Microsoft tool, I imagine. Yes. Yeah. No, actually it was just like I mean I'm I live in GitHub and GitHub copilot. So it just like was like okay let's go build this Chrome extension. Yeah. Are you releasing this for the general public? No I mean this this that's the fun that's the ma amazing thing. It took me like 10 minutes to do this. I Okay, let's link to it. Let's get it out there. Open source this thing. Okay.

you mentioned Satia. I have a question about this. So you're one of the very few people that have worked very closely with both Satia and Sundar at Google. Let me ask you this. How do their leadership styles differ? And is there just like a fun story you could share about each of them? Yeah, I do feel I do feel lucky to have you know kind of have a window into these two amazing leaders of this generation. I would say I mean again no surprise they're as you'd expect from CEOs of multi-trillion dollar market cap tech companies.

They are 99.99 percentile in like almost every dimension you'd think of, right? intellect, empathy, leadership, you know, be product strategy. there are of course flavors of differences. I was at the technical adviser for Sundar with the first at at Google and set up kind of the office of the CEO there. and there again a matter of like time and context because a lot there's a lot more consumer oriented focus there. So what I did find so great at added is being really calm and measured and thoughtful in terms of u you know taking making sure that things

have dealing with the complex ecosystems right if you think about the phone ecosystem or even like the search and publisher and advertiser ecosystem it's a very complex ecosystem he was a master at that he's a master at that and I think on Satya I find it amazing the appetite he has for learning and fine-tuning his mental models and just like the the zoom levels that he can operate at. the macro, the strategy, what's the game, but also the micro, hey, what why are we like here's like a specific insight that I saw on Twitter and like you can count on the fact that he's ahead of pretty much everybody else in terms of spotting those early things too. So, it's it's just been like like you know, learning from the fire hose as as they put it. What a cool opportunity to work with two incredible folks. Okay, let's go in a in a whole different direction. Let me just ask you this question that I've been asking people more and more. What's the most counterintuitive lesson that you've learned about building products that goes against common startup wisdom, common product building wisdom? I don't know if it's a I mean as common as it should be and it's like a counterintuitive thing but I've repeatedly learned that when you're doing something new Oero to one the temptation is to kind of think about you know it's like that South Park episode one think about the problem step two underpants is underpants exactly right so I do feel like there's a temptation to rush and say to go to scale scale before solve. So I've always said to my teams solve before scale, right? So what that what that does mean is there's a different posture and different mode when you're trying to solve a problem versus scaling something that's either postp productduct market fit or even at least like in this roughly in the ball ballpark. So to give you a couple of examples, right? I think when we when you look at the solve stage, there are wide lurches. You got to be very comfortable with the fact that you're day one thinking about hey a plant detection tool and then day 15 you're like oh actually the tech is really good for translating you know foreign language by the way this is not hypothetical this is what we kind of like looked at in Google lens back back then and said okay like where what is the intersection and so on so from the outside it looks like chaos but actually in the and you should be very com not only tolerant I think you should be like should have an appetite for that because the last thing you want is prematurely like you know fix on one local hill and then you're climbing that and startups and entire product areas and companies big companies make that mistake and 3 years later you're like oh how do I get off this hill so I'd say that's one big comprehensive thing like when you're trying to think about what mode you're in are you in the solve mode are you in the scale mode one example is kind of making sure that you're comfortable with the chaos I think the other lesson I've learned learned is the danger of metrics, right? And I think again if you have worked on you know rule search or if you worked on you know like office products you really have like a very fine grain sense of what are the metrics for this product you have the input metrics out you have the whole shebang but when you're looking at something Oero to one if you decide on a metric too prematurely that's false precision first of all right like you kind of u I mean CTR when you have like thousand people doesn't mean anything. you know retention also may not mean anything. So really being very wary of like this big guy big girl of grown-up metrics as I call it right you are looking for more qualitative the sound of click and what is your as the other an you know kind of the handler uses what is your set timer and play music right so if you look at like Alexa and like Siri and Google assistant and all these things they had a very promising broad interface you could say anything but I think there was one or two things that it was really good at, right? Like you could set a timer, you could play music, and you could play trivia. And so, you got to nail those things before you say, "Oh, yeah, here you can do anything with it." Which is not a good recipe. That's exactly what I use my Google Home for.

so basic. I don't do the trivia thing though. Maybe I got to give this a shot. Got to play that. Yeah. There's something along these lines that I've also seen you talk about, which is how to go zero to one with something.

just kind of little framework for helping you know if this is the right time for this idea. How do you think about that? Yeah. I and when we when you think about the solve mode and this is again like sticking with my whole you know living in one year in the future I I gravitate towards the Oero to one and solve mode products completely thinking about new category of products and what I found about the hard way I would say is that you do want to look for at least two out of these three factors inflection points here if you want to make a really good product. Number one is there a shift is a step function in the tech right that's somewhat obvious I would say like you know deep learning was one for Google lens back then speech recognition was a step function for like conversational search I would say for Robin Hood you know the the generational shift was very clearly and the fact that phones were a primary means for you know you could actually have an app mobile app for finance that you could use. So look for that inflection right what is the tech inflection and right now of course like NLMs and reasoning models are that step function but that's not enough I would say the second factor that we should look for is what is the consumer behavior shift right so to give you an example when we started working on Google lens what we said is look people were taking mostly pictures for sharing right selfies and sunsets and so on and suddenly when storage became free and mostly free and everybody had phones everywhere all all the time. You took pictures of everything, right? And then you had like enough of pictures or you used the camera as the as the keyboard for your for your world, right? For the real world. And so how do you kind of then say, "Oh, this consumer shift is big." And so therefore kind of like as it as you go order of magnitude more photos, then you want more to come out of them. And you can apply AI to that.

And I'd say the third inflection point particularly I would say in enterprise but also in consumer is the business model shift right how do you is there an inflection point natural inflection point in the business model so any great products if you think about like you know all the way from search again like the the second price auction and the fact that you had like you know CPCs same thing with SAS and the fact that you could actually charge or monetize enterprise products in a different way And with AI of course like the monetization is a whole different like I mean you've we've just barely scratched the surface of whether you do you know seat monetization usage like on tap and then of course outcome based stuff outcome based monetization hey have you solved the problem for me and then I will pay you some fees right so all three like to me are you know kind of like great but at least two out of three for a good product so this essentially when investors at startups, they're always asking why now? Why is this the time to start this thing? And so your advice here is you should there's three ways to look at it and you should two two of these three should be true. There should be a a shift in technology some new technology that has enabled this now recently. There's a shift in consumer behavior and then there's maybe a new sort of or you've invented a new business model like any way to monetize something that it gives you an advantage over folks trying to do it today. Absolutely. Awesome. And you didn't mention Robin Hood, I think, in that example. That was another good example of Yeah. Yeah. I mean, talk about the business model of kind of again like not having a zero u you know, zero fees, right? And again, like the combination of all of these things is what can unlock it. Not you can't just say, oh, we'll just have a much much more better intuitive interface and hope that you know, people will switch to it. Okay. So speaking of Oero to1 products, I'm going to take us to a occasional segment on this podcast that I call hot seat corner and I have a question for you that is on my mind and it's come up in a couple recent podcast actually. So there's these companies like cursor vzero lovable bolt replet that are like the fastest growing companies history. I just saw that cursor hit 300 million ARR in two years.

Interestingly you guys were very well positioned to do really well in this space this AI coding tool space. You guys said co-pilot the first tool in the world at this stuff. So ahead of everyone you build VS code which is what all these companies are forking to build on. You have incredible AI infrastructure, incredible AI talent. So this could have been your market. What happened? What happened to partner? You know it's interesting the framing, so I'm a daily user of GitHub copilot.

and I would say look the the if you unpack I think the thing the the beauty of this is that code generation is become an amazing tool that LLMs have unlocked right so it is not so it is actually really good excitement and action that now code generation has just opened up all of these things that we talked about the whole idea of like prototyping go from idea to mocks an idea to kind of a clickable prototype in like in a few minutes those are the kinds kinds of things that of course we should expect code generation to enable the way I think about you know how we we are positioned and like what we what we do with GitHub is so it's a system not just a product or a set of features if I think about GitHub it's for folks who are who have the repo there right and you have kind of of course you have the assistance in terms of autocomplete and you can chat but now we have the agent mode it's one one of the fastest you know kind of loops that we are seeing really strong positive feedback. So in some sense when you have a system what you are looking for in terms of building and designing it is not just a single product that can go grow but it's the what is the repository what is your context what are the set of features that grow from your expertise right if you're a really expert coder you want kind of like the you know assistance the this product needs to scale for that if you're a w coder you should still be able to do that and so on right so that I think is the way that that GitHub is positioned to build on and like growing honestly really well. That's so interesting. So bas like the core of this is everyone ends up in GitHub anyway no matter what tool they use and that's kind of the yeah and I think the yeah the idea again is that you know code generation as a tool we'll unlock a lot more products. I mean they're not all competitors to the fact of u they're not all kind of you know doing the same job. I think when you're at the end of the day like you're building code for companies to run on, you need to have a system, you need to have kind of the ability an entire Swiss Army tool toolkit, right? Not just the autocomplete, not just a chat, not just like a software agent that runs and you kind of like handhold. You need all of this to work together and that's what the GitHub product is going after. All roads lead to GitHub. on the flip side of this question, there have been probably 5,000 startups that have tried to disrupt Excel and you guys just keep winning. So something there is working really well.

That is so interesting you say that. So when I came to Microsoft and I'm an Excel fan, so I actually had a conversation with one of the OG Excel product folks. I was like, man, what is it about this this product? And he said a couple things that were really interesting for me that just stuck with me. One is you know I said hey you know Excel is a proof that non-coders also have to program right programming is really powerful and it's the tool that gives all of the non-coders really powerful programming you know ability and I thought that was just like really u striking and then the second thing that I found out super cool did I don't know if you know this but I didn't know at least before two years ago that there are these amazing Excel championships like world XL championships where you see folks who can do just magic and to me I think the insight here is also that some tools are harder to learn perhaps in the beginning there's friction in terms of learning but great to use right so it's a very good case of hey the learning curve initially the onetime learning curve might be tricky but it is because there's so much power and depth in the in the tool That's so interesting. I never thought of Excel as a programming language, but but it makes sense. And I feel like once you get used to it and this is just the way

things work, you're kind of stuck there and everything else has to basically copy that model, which is hard to be as good. Yeah. And I think the the depth and the attention that the that the team has given and again that's the compounding effect over you know decades of working on like deep deep signal right from people who live who depend on it day in and day out. Yeah. Okay. to to kind of start to close out our conversation. I want to ask this question around your career. I find that most people have a like one moment in their career that changes the trajectory of their career. It could be like a manager they had. It could be a project they worked on. Could be just a job they landed. What would you say is the most pivotal moment in your career that eventually led you to becoming chief product officer at Microsoft?

Actually, there is one moment where you know it was a turning point for me. I was in Google search. I was working on this idea that I thought should just work and it didn't, right? Like I I I said, "Hey, these phones are becoming a thing. Personalization has to be important." So I I probably banged my head against the wall for a year or so. trying to make personalization work. And it turns out when you have a you know query that you put into Google search like the personalization didn't matter as much. And so you know we disbanded the team. But then I think I started working on this product called Google Now which was a twist on that which said hey actually on the phone we should be able to like push content.

It's not about like you know searching with personalization. For example if you have a flight coming up it you should be able to say hey connect the dots and say you should leave now for the you know given the traffic and where where you need to go and so on. or if you're deeply interested in I don't know standup comedy with deadpan artists you should check out Mitch Hedber like these are kind of like these really moments that the smartphone should be smarter so I I let that product through the kind of the initial Oero to one phase and that was a pivotal moment it made me realize two things one I really love seeing around the corner and kind of seeing where where things go and building the product to rise to the occasion way more than you you know the the scaling and s sustaining products second it's harsh but being early is the same as being wrong you know this is prel's pre-deep learning a lot of the really amazing ideas in terms of next token predictor etc we we've been thinking of it but you know didn't have the horsepower to go the interface was great the intelligence wasn't there and I'd say the third thing that stuck with me is I got to work with some really smart like they talk about talent density you Right. And I think really smart people have gone on to do like amazing things.

And so kind of like it gave me a taste of what a small group of people can do. It's such a great story because because it didn't work out right in the end like Google now kind of went away, right? And by the way, I super remember that product. It was very cool. I remember looking at it as very like delightful and happy. And so I I also have this segment on the podcast called Failure Corner where people share a story of failure and how that helped them. And I love this as a combination of those two.

Yeah. I mean I I'm not going to lie. I think it was it was u it's painful when you do that because you you see the vision of what can be and what is and sometimes it's hard limitations. Sometimes it takes like you know in this case it takes 5 years or 10 years to kind of like really unlock the intelligence but sometimes it's a it's one or two key clicks click stops away from the product being great. and part of figuring out is knowing when when you're in what situation. How long was that period from from starting out until just like moving on and it's not

working? Yeah, I would say in that case one of the good things is again like the the it led the foundation of it was one of the foundations of the Google assistant and of course as the LLM's you know step function happened now with Gemini it kind of like works out. and I think it's the same thing across the board which is sometimes you want to kind of figure out the invariants that do work right that can then then go on to the next version of the product and other times you just have to start over is Google now the first agent before agents that's what it feels like that was certainly the idea you know inter but it is fascinating to me that the interface that there we had the opposite problem like whether you think about all the voice assistants right the interface is like we overshot and the intelligence wasn't there. Today I feel like there's an opposite problem. I think these these things have amazing intelligence and the interface we have largely is like the aol dialup modern chat bot. We've covered a lot of ground.

Is there anything that you wanted to chat about or leave listeners with maybe a last nugget of wisdom before we get to our very exciting lightning round? I think I would say one thing that I'm really excited about is this idea of figuring out how we as people and agents collaborate together. Right? I think there's like some great set of products and experiences to be reimaged. That's my other Roman empire which is how do we actually have this co-working space where you know you have kind of like the the humans and agents and how do you actually kind of have an output that's much much more significant than what any one of us or any few of us can produce. Well, I need to hear more about this. What do you when do you imagine a co-working space of humans and agents? What does this look like? Is this like Microsoft Teams or is this like a physical place with little robots? Oh, I hadn't thought of the physical place, but I I I I am think I am thinking a lot about kind of, you know, right now all of these experiences are very single player, right? And I do think there's an opportunity to think about how do we again I'm living one year in the future. How do we actually have like you know collaborate with each other but with also with agents and really figure out for example what tasks can we delegate? What can we kind of like inspect? How do we actually have information that flows between people that agents can mediate and so on?

All right, I'm curious to see what you guys got cooking. With that, we've reached our very exciting lightning round. Are you ready? Let's do it. Let's do it. First question, what are two or three books that you find yourself recommending most to other people? Oh, I have recency bias, but I've been reading this book called the brief history of intelligence. phenomenal book and you know like lots of lots of underlining for me and I think it kind of the premise is to it it looks at the evolution of intelligence like human intelligence and kind of the the brain development and connects that to what we're what we're seeing with the eye.

Do you have a favorite recent movie or TV show that you really enjoyed? Hacks. I've been watching this. It's about a woman who who's a like a great stand-up comedian of I think it's set in kind of like the the fact that she she grew up I think in the '7s and ' 80s and kind of like really tried to break through in an industry that hasn't traditionally been like very friendly to women. So really fun and quirky.

Do you have a favorite product that you've recently discovered that you really love? Could be an app, could be some physical. I I do use a lot of Microsoft products. GitHub Copilot being one of them. But I think the one that I maybe I'll pick is Granola I think is the name of the app. I I found it really useful. I just gave it a spin the other

day and I'm like oh this is really useful in terms of being able to u you know again like without being intrusive just just capture the thoughts notes and structure it put some it it it felt like one of those things where yep the confidence of a few things like we were talking about right like the transcription real-time transcription tech has gotten really good voice recognition is great and then enough of the LLM magic on top of it to kind of make it structured and contextual.

I am a huge fan of granola. I'll give a quick pitch here. If you become an annual subscriber of my newsletter, you get a year free of granola for your entire company. Did not know that. There we go. So, and then just check that out. Lenny's.com and you click the word bundle and you'll see how to do that. Very cool. Very cool. two more questions. Do you have a favorite life motto that you often come back to when you're dealing with something maybe you share with folks that they they find useful as well in work or in life? I have one. In fact, actually this is my email signature for I don't know for the last 20 years or so. it says the best way to predict the future is to invent it. I think it's a quote by Alan K. I find it useful for two things. One is you know no one knows anything like when you think about like all the folks who are program you know kind of think about hey this is the this is exactly how everything is going to look and this is exactly the sequence and so on. I think there's no substitute to experientially like building it. And and I think the second part is, you know, like if you think there's something that is that should exist, go build it. I love that. Final question.

We've talked about standup comedy a bit. Is there a is there like a favorite under the radar stand-up comedian that you think people should go check out? Oh, there's a there's a couple of them. So, one I think there's a there's an Indian-American or I think I think a British Indian stand-up comedian. Her name is Sindu Vi. Super smart, you know, mom comedy. And I think the other one that he this is definitely not under the radar, but like I've just like love his stick is Nate Badzi.

He's just so good. Perna, this was amazing. Two final questions. Where can folks find you online if they want to reach out maybe and follow up on anything you shared and how can listeners be useful to you? You can find me on LinkedIn and Twitter. Aperna CD is the handle. I do post stuff a lot more on LinkedIn these days. So you know would love would love to hear thoughts, comments, conversations there. I'd say one thing that would be super interesting is if any of this stuff spark conversations particularly around like kind of you know this what do what can a small team with a lot of AI tools do or new products that folks are really excited about saying that they should exist. Hit me up.

Amazing. Pera, thank you so much for being here. Thank you. Bye everyone. Thank you so much for listening. If you found this valuable, you can subscribe to the show on Apple Podcasts, Spotify, or your favorite podcast app. Also, please consider giving us a rating or leaving a review as that really helps other listeners find the podcast. You can find all past episodes or learn more about the show at lennispodcast.com. See you in the next episode.