

Y COMBINATOR

Startup School India

Notes on the talks

Harshil Mathur · Aadit Palicha · Varun Vummadi
Mukund Jha · Vidit Aatrey · Lalit Keshre
Closing panel



The seven sessions, in the order they were held

One chapter per session, in the order the sessions took place. Each chapter opens with the video's thumbnail and a link to the talk. Written from the transcripts, and kept to what each speaker actually argued, in plain language, with no buzzwords.

Released 6 May to 27 June 2026.

Source playlist: youtube.com/playlist?list=PLQ-uHSnFig5M5RuOSM11oH-EjjQpkFell

Contents

How to read these notes	3
1 Harshil Mathur: AI Is Compressing Every Moat	4
1.1 From an oil job to payments	4
1.1.1 The wrong first market	5
1.2 Why the hard part became the advantage	5
1.3 The week the bank pulled the plug	6
1.4 Turning down the acquirers	6
1.5 Growing fast while barely spending	7
1.5.1 Being small as an advantage	7
1.6 Refusing to behave like the incumbent	7
1.7 The mistake he names in himself	8
2 Aadit Palicha: Starting From The Extreme	9
2.1 A WhatsApp group in a Mumbai neighborhood	9
2.1.1 On dropping out, with a less heroic answer	9
2.2 Entering a market that already had giants	10
2.2.1 The pivot, which began in an apartment	10
2.2.2 Why being young and unaware helped	11
2.3 What the company actually is	11
2.3.1 Where it goes	11
2.4 Where AI actually landed	12
3 Varun Vummadi: Turning Down The Safe Offer	13
3.1 What the company does	13
3.2 A small town, then Kaggle, then a rejection that worked	13
3.3 Two pivots, both driven by customers	14
3.3.1 Beating a much larger competitor	14
3.4 What he thinks agent products actually reduce to	15
3.5 Running the company on automation	15
3.5.1 A mistake he names about himself	16
3.6 The decision his family disagreed with	16
3.6.1 On ideas, and on charging	16
4 Mukund Jha: Living At The Edge	18
4.1 Nine months to \$100 million	18
4.2 Five startups, and a habit of following intuition	19
4.2.1 What the previous company taught him	19
4.3 Six months of tinkering with no objective	19
4.3.1 Being rejected for the ambitious version	20
4.4 What is actually under the product	20
4.4.1 Why they attacked a benchmark	21
4.5 Not being first, and winning anyway	21

5	Vidit Aatrey: Rigid On The Problem, Flexible On The Solution	23
5.1	The scale, and the gap it is aimed at	23
5.2	The observation that started it	24
5.3	Version one, and a mistake he calls stupid	24
5.3.1	Version two: observe rather than ask	24
5.4	Finding the power users	25
5.4.1	Why WhatsApp was the right surface at the time	25
5.5	Killing a working business	25
5.6	The next version, aimed at the same gap	26
6	Lalit Keshre: Love It Or Hate It	28
6.1	Third startup, and the first idea failed too	28
6.1.1	The idea that did not work	29
6.2	Opening everything up, with no way to make money	29
6.2.1	Four years with no revenue	29
6.3	How they actually talked to customers	29
6.4	Choosing to be regulated on purpose	30
6.5	Being the power user of your own product	30
6.5.1	On using the tools himself	31
6.6	Four co-founders who stayed aligned	31
6.6.1	What is next	32
7	Closing Panel: India Can Create The Largest AI Companies	33
7.1	A note on this chapter	33
7.2	Why this wave is different from the last one	34
7.2.1	Selling to a market you have never visited	34
7.3	On advice, and why the traditional kind may not apply	34
7.3.1	How you develop an independent point of view	35
7.4	Why the founders keep getting younger	35
7.4.1	Second mover advantage	35
7.5	How far you can push the tooling	36
7.5.1	And for people who cannot afford that	36
7.6	What they actually look for	37
7.6.1	What has not changed	37

How to read these notes

Six founders and a closing panel, released over about seven weeks. Five of the six companies serve mass-market India, which gives this set a different centre of gravity from a conference about frontier technology. Four things recur:

1. **Almost nobody's real business was their first idea.** Razorpay went to YC selling to schools. Zepto was a WhatsApp group under another name. Giga was an education product, then a fine-tuning company. Meesho was a local-shop marketplace, then a WhatsApp toolkit, then a supplier. Groww was an automated adviser. Emergent was a testing agent. The closing panel makes this explicit: a good idea is usually not obvious, and you have to be in the field building something to notice it.
2. **The hard part is the moat, but only one kind of hard counts.** Mathur draws the distinction most sharply: if it is hard to *sell*, that is a problem; if it is hard for other reasons, that keeps competitors out. Jha chose the last mile at Dunzo and the whole of software engineering at Emergent. Keshre deliberately chose to operate inside regulation, because a strong choice removes variables.
3. **Start from the customer, then work backwards to what is possible.** Palicha states it as a method: remove the constraints, ask what the most extreme good experience would be, and work back. Aatrey's whole company is the same move applied repeatedly, and his one named mistake was talking to sellers and never to consumers. Keshre spends two hours a day in his own product and two more talking to customers. Mathur's rule is that support is a trust channel rather than an efficiency problem.
4. **Being small, young or unaware is treated as an asset.** Razorpay supported a new payment system before the largest banks because they had nothing to lose, and gained a six-month lead. Zepto's founders credit not knowing how hard it was. Giga won a very large customer with eight people. The panel's version: you are no longer limited by your ability to build, only by how fast you can learn, which favours the young.

Two notes on sourcing. The closing panel's two guests introduce themselves by name and are named here; the YC partners on stage are not consistently identified in the captions, so their points are attributed to the panel rather than to individuals. And these notes are written from automatic transcripts, so wording is paraphrase throughout rather than direct quotation, and a few names are corrected against the video thumbnails where the captions garbled them.

Boxes are used consistently throughout:

- **The one thing.** The point worth remembering if you remember nothing else.
- **Against the grain.** A belief the speaker held that the people around them did not.
- **Try this.** Something concrete enough to actually go do.
- **Watch out.** A trap the speaker explicitly warned about.
- **Parting advice.** The closing thought each speaker left the room with.

CHAPTER 1

Harshil Mathur: AI Is Compressing Every Moat



"Harshil Mathur: AI Is Compressing Every Moat"

Harshil Mathur CEO, Razorpay. Interviewed by a YC partner, 6 May 2026.

Video: <https://www.youtube.com/watch?v=X5bABLCulHA>

1.1 From an oil job to payments

Razorpay was the first Indian company Y Combinator invested in, and today processes over \$180 billion a year. None of that was the plan.

He was a coder from school days who knew nothing about finance. After graduating he was placed at an oil company in the Middle East, never enjoyed it, and spent his evenings and weekends building things. One of those side projects was a crowdfunding platform, and it needed to accept payments.

Going through the standard channels was hard, and the experience was poor, because the existing systems were built for large companies rather than for a developer wanting to start. His reaction was the classic one: how hard could this be? He calls that a dumb question in hindsight.

THE ONE THING

What turned it from an annoyance into a company was checking whether it was just him. He went into startup groups on social media to see what others were doing and found nearly everyone had the same problem. It was easier to accept cash in India than digital payments, which as a technical person made no sense to him.

His framing of why that mattered: the point of technology has been democratization. Social media gives everyone a voice, the internet gives everyone a channel, e-commerce gives everyone a place to sell. When the digital system is *harder* than the physical one, it does the reverse. It creates silos.

1.1.1 The wrong first market

They needed a way in, and reasoned their way to education. An enormous volume of school and university fees was collected through non-digital channels, so digitizing even a fraction looked like a large opportunity. They applied to YC saying they would sell to educational institutions.

Then he actually went and sold. He remembers visiting small universities and being asked why they should accept fees digitally. He explained it would cost them roughly one percent. The reply was: then why not just charge one percent more in fees?

WATCH OUT

That is when he understood his customer did not care. Education in India was a market where the institution could dictate the payment method and students would comply, so nobody wanted smooth collections. The logic of the market size was sound and the customer motivation was absent.

Meanwhile, because they worked in a co-working space, startups around them kept asking to use what they had built. That segment was small but growing for them, so they pivoted away from education entirely. He considers it one of their better decisions.

1.2 Why the hard part became the advantage

This is the most transferable argument in the talk. They spent the whole three months of YC without doing a single live transaction, because they were waiting on approvals. It took about a year after that to process their first one. Licences, certifications, approvals, then final approval.

Very few technology businesses have that kind of waiting period, and the standard advice is to go and start selling. They could not sell.

AGAINST THE GRAIN

What he kept telling the team: yes, there are a lot of hurdles, and everyone who comes after us faces exactly the same ones. So the hurdles become the moat.

His argument that regulation is fairer than it looks: no company, however large, gets to skip the guidelines, the rules, or the year-long process. And very few have the patience or energy to go through it. That is why their space still has far fewer competitors than most well-funded sectors in India. You do not get a hundred new entrants in a month, because it is not easy to start.

He is careful to separate two kinds of hard, and the distinction is worth keeping:

- If it is hard to *sell* to customers, that is a problem.
- If it is hard for other reasons, that is not a problem. It is a moat, because it keeps others out, and if your customers genuinely care, they will love you once you build it.

He is also honest that conviction was not steady. He says it is easy to claim in hindsight; in reality he sat with his co-founder most months asking why they were building in this space when companies around them were getting funded in e-commerce and food delivery, and they were coders who could build something else. What held them was that every potential customer they met confirmed the problem and confirmed nobody was solving it.

1.3 The week the bank pulled the plug

Their near-death moment, and the source of a lasting operating rule.

They got final approval a week before demo day, having previously considered delaying to the next one. They launched, did demo day, and investors were interested. About two weeks later the bank that had enabled them withdrew support after a complaint, and shut the platform off. Around 50 live merchants were suddenly unable to collect money.

In a payments business that is close to the worst possible position, because someone has just trusted a small provider with their revenue. They tried pleading with the bank and got nowhere.

TRY THIS

The principle they set: call every single customer and tell them exactly what was happening, why, and what was being done about it. His observation is that when a crisis hits, people stop picking up the phone, because there is nothing good to say. Six of them sat in a room and called everyone. Plenty of customers were furious, and some of the ones who abused them down the phone are still customers today.

It took four or five days to get another partner live. But the durable lesson was about what support is actually for.

THE ONE THING

His rule today: if a support conversation goes past three or four exchanges, you pick up the phone. Not because it is efficient, but because a human saying they are on it builds trust that efficiency cannot.

Which leads to a deliberate exception in a company that is otherwise aggressive about automation. They use AI in everything *except* replacing customer support, because in their business support is not primarily about solving the problem. It is a channel for demonstrating that a person is spending their time on you. B2B, and financial services in particular, is a business of trust.

1.4 Turning down the acquirers

Offers arrived early, before they had even launched, because YC had made them visible. Some of the largest payment companies in the world suggested they join or build India on their behalf, and he found several genuinely exciting.

What decided it was a disagreement about the size of the future. His concern was not the present but the investment India would need two years out.

THE ONE THING

The numbers make his point better than the argument does. In his YC application he wrote that India's payments market was \$60 billion. Razorpay alone now does \$180 billion, and he still thinks that is small against a market he expects to exceed a trillion.

Which creates an impossible pitch. Telling a global company that a \$60 billion market needs \$500 million of investment is very hard, unless they already understand that the market will scale enormously. Most markets in the world do not grow at that pace, so that understanding was

missing. His conclusion: unless a partner gets you to your goal faster, there is no reason to take one. They never found one.

1.5 Growing fast while barely spending

Between roughly 2017 and 2020 they grew enormously while being unusually capital efficient, which at the time was not considered a virtue.

They raised around \$10 to \$11 million in their Series A and had a burn under \$200,000 a month, so they put the money in fixed deposits. The interest exceeded the burn, which made them profitable and made their investor unhappy, since the money was given to them to spend on growth. He had to explain himself in what he calls a hard conversation.

AGAINST THE GRAIN

His reasoning is a clean argument about why B2B differs from consumer. In consumer you spend to acquire, spend to grow the customer, spend to build engagement, and monetize later through some other channel. In B2B you add value to a business and the business pays you for it. There is no point burning money to engage someone whose decision is already logical.

The flip side, which he does not soften: your value is only what you deliver today. If you stop adding value, the customer leaves tomorrow. That makes the business more rational and more intense, because the person on the other side is re-evaluating every day whether you are worth more than you charge. Easy to say, harder to compound as the organization grows.

1.5.1 Being small as an advantage

The best illustration of the whole talk. When the national instant payment system launched in 2016, there were many skeptics, and the country's two largest banks had not integrated with it. So most payment providers did not support it, reasoning that without those banks nobody would use it.

Razorpay had 10,000 merchants and nothing to lose. They became the first payment gateway in the country to support it, before the largest banks were live. Then the large banks went live, then demonetization happened, and suddenly everybody wanted it. No other provider was ready at that scale, so several of the largest consumer companies in the country moved onto Razorpay within a couple of weeks, having previously served only smaller businesses.

THE ONE THING

His framing of the bet at the time: the worst case is that the new system does not take off, and they are fighting the same battle they were already fighting. The upside is a unique wedge. It took six months for any competitor to catch up, and in that window they entered markets they could not otherwise have entered.

1.6 Refusing to behave like the incumbent

His answer on AI is about organizational courage rather than technology, and it is the sharpest part of the interview.

It starts with him personally. He is candid about being deep in coding tools again, and says the reason it matters is not fun: when you spend your own time in the tool, you understand what it can actually do, and that lets you make fundamental decisions. He also makes a nice aside that most founders start companies because they love building, and then the layers accumulate and they become people managers, which nobody signed up for. These tools let you get back to building.

TRY THIS

The exercise they ran a couple of months earlier: if I were starting Razorpay today, how would I build it? They put the whole thing on a whiteboard. How would integration look, onboarding, support, the platform, how people would interact with us. Then they said: make that happen. They rebuilt the platform end to end and launched it.

WATCH OUT

His name for the trap is the incumbent fallacy: believing that you have everything, you are well set, and if something changes you will respond. The market changes faster than you can respond, so if you start responding when the market has already begun changing, you are too late. Any company whose plan is to respond to the market is, in his words, already dead.

He does not pretend this is comfortable. It means telling teams with large budgets and established work that you will hurt yourselves in the short term to pivot to what makes sense long term. Every team has that fallacy. He says it is the only way to survive.

THE ONE THING

The claim the video is named for. Technology used to be the moat, then it became how fast you can execute, because the capital moat and the infrastructure moat both came down. He expects building itself to stop being a moat too, because building will compress massively. What remains is deciding *what* to build, and how fast you can move on that decision.

1.7 The mistake he names in himself

Asked how he has changed, he goes straight to an error rather than a strength.

The pattern: you are in the trenches for the first two or three years, the company takes off, you hire excellent executives, and you conclude you can simply manage the leaders. He did that for a couple of years and calls it the worst mistake you can make.

He is careful about the line. You do not want to micromanage everything. But on the things that genuinely matter, the founder has to be in it. His example is product vision and direction, which no leader will think about the way a founder does. He had genuinely strong leaders, and the thing that differentiates a company is a conviction the day-to-day execution layer cannot supply.

THE ONE THING

The underlying point, which he says will never change: nobody is going to care about your company as much as you do. Not on day one, not in year ten, not in year twenty. It took him about ten years to fully accept that he needs to be in the core things and can delegate everything else.

PARTING ADVICE

He pushes back on the idea that easier tools make entrepreneurship easier. The tools required to build a company are getting easier; the core of being an entrepreneur is connecting deeply with a problem and spending all your time on it, and that is unchanged.

His specific worry is that because building is easy, it becomes easy to latch onto problems you do not actually care about. You can deploy something today. So the question to ask is not whether you can build it. It is whether there is a problem you could spend the next ten years of your life on, because that is what building a company still takes.

CHAPTER 2

Aadit Palicha: Starting From The Extreme



“Zepto: How Two 17-Year-Olds Built India’s Largest Seller Of Fruits and Vegetables”

Aadit Palicha CEO, Zepto. Interviewed by his former YC group partner, 19 May 2026.

Video: <https://www.youtube.com/watch?v=YKZCU0ynEbs>

2.1 A WhatsApp group in a Mumbai neighborhood

He and his co-founder started working on it at 17, having watched Startup School videos in a bedroom. They had applied to study in California, got in, and then the pandemic put them back in Mumbai with nothing to do and no reason to attend online classes. So they took a year off to work on a project.

The problem in front of them was immediate: during the first wave, people could not get groceries. Small local shops had labor problems and the larger next-day services were backlogged. So they started a WhatsApp group and delivered for their neighbors themselves. That became an app under a different name, and eventually became Zepto.

2.1.1 On dropping out, with a less heroic answer

Asked about the decision to leave a place at Stanford two or three months before starting, he deliberately declines the dramatic version.

THE ONE THING

They treated it tactically rather than as a leap of faith. They took a year, and in that year got to an early sign of product-market fit at around eight or nine months, when they could see retention and compounding. Only once they were doing roughly 10,000 orders a day did they conclude it was a real business worth committing to.

So his advice to people asking whether to drop out is the opposite of the romantic version: you should have a real product that is working, and you should not take the risk before you have the proof of concept. In their case they also had investor interest by then, so they had a term sheet, a working business, and customers who liked it.

2.2 Entering a market that already had giants

Grocery delivery was not a new idea in India. Several billion-dollar companies were already operating at scale, and people had been chasing the opportunity for twenty years including offline.

His answer to why that did not deter them is that they never framed it as a competitive choice at all. It was bottom-up. Because he and his co-founder did most of the early deliveries themselves, they would hand over a bag of groceries and ask for two minutes of conversation at the doorstep: why did you use us, why not one of the others, why not do it yourself?

THE ONE THING

What they learned was that customers were not satisfied by anyone, including Zepto at that point. He is explicit that their first version had not found product-market fit either: the selection was incomplete, the pricing too high, and the delivery times outside their control.

Reduced to four things customers actually care about: speed, quality, selection, and price. The question became how to push all four at once.

2.2.1 The pivot, which began in an apartment

Delivering from existing shops meant no control over the customer experience or the delivery time, which meant they could not fix the problems customers were describing. So they bought stock and put it in his co-founder's apartment. That was the first small warehouse.

The response in that one small area was strong enough that they built a cardboard version of a warehouse in another neighborhood, then a proper one.

THE ONE THING

The number that made the decision for them. They expected incremental improvement across delivery time, quality, selection and pricing. What they saw was a tenfold difference. The single neighborhood served from their own warehouse did three or four times the volume of the entire rest of the city served from local shops.

AGAINST THE GRAIN

When they started building their own warehouses, plenty of people said it was the wrong approach. His diagnosis of why: those people were reasoning backwards from supply chain constraints or financial targets, deciding what was viable and then offering the customer the best that allowed.

Zepto reasoned from the opposite end. Remove the constraints, ignore the laws of physics, and ask what the most extreme positive customer experience would be. Then work backwards to how you make that possible at scale. That is where ten-minute delivery came from. The customer was telling them it made sense while the commentators were telling them it did not.

The consequence he says is impossible to forecast in advance: because they got the experience right, they got far more throughput per warehouse than anyone expected, and at high volume in the same footprint the unit costs fall by themselves. When people are pulling the product out of your hand rather than being sold to, the economics improve as a side effect.

THE ONE THING

His conclusion, stated flatly: it is impossible to build a financially viable business without customer delight. Customer delight is where financial value starts.

2.2.2 Why being young and unaware helped

The interviewer credits him with unusual instincts; he redirects the credit and points out the failures being skipped, including a period during the batch when their group was the worst at getting tasks done.

His actual explanation is about noise. They were extremely young and naive, and the advantage of that is not knowing how difficult the thing is, or the many ways people have tried and failed before. Being locked down cut out the noise of startups, investors, and advice entirely.

THE ONE THING

What was left was them and the neighbor down the road who wanted her groceries. So the only question they had was how to get the thirty or forty people around them to order, and there was nowhere to hide: if a customer was unhappy it was directly in front of them. He connects it to the standard advice about making a hundred users love the product rather than chasing thousands.

2.3 What the company actually is

Most people experience Zepto as an app. He is insistent that it is a logistics and retail business. His own description, stripped of software and warehouses, is that they sell everyday groceries, and that customers think of them as where they get their staples rather than as a technology company. He says that clarity is useful.

Underneath the app: hundreds of small design decisions inside each warehouse, covering selection, layout, restocking algorithms, trucking, and task management. Over 200,000 people employed including delivery partners, pickers, packers and drivers. Industrial automation on the back end so goods move from large suppliers through to the warehouses with as few people involved as possible. A meaningful part of their technology team now works on hardware.

THE ONE THING

His reason for all of it is not efficiency for its own sake. Every unit of cost saved across the supply chain is either given back to the customer or reinvested in better selection or faster delivery. The chain of reasoning starts at the customer and ends at the robotics.

They also run what he describes as the country's largest fruit and vegetable supply chain on the platform, sourcing millions of units a week from farmers.

2.3.1 Where it goes

The vision he describes is grocery infrastructure for the country: an organising force in a supply chain that is far less organised than its equivalents elsewhere, at lower cost and with better quality control, across the top forty or fifty cities.

Two second-order effects he draws out. Quick commerce platforms have become a route to market for new consumer brands, which he compares to what large retailers enabled elsewhere decades ago,

and he would like to see more large consumer brands built on top of them. And the employment created on the ground, which he treats as a straightforwardly good thing.

He also notes the advertising business, where large brands bid on search keywords, has gone from inconsequential two years ago to hundreds of millions in annual revenue.

2.4 Where AI actually landed

Concrete rather than aspirational, split across three areas:

- **Supply chain.** Forecasting that used to take a team of people days now runs across millions of units a day with no humans in the loop. The benefit is not headcount, it is agility: higher forecast accuracy means faster dispatch and more throughput from the same space.
- **Advertising.** Tools that help brands generate keywords and predict which will give the best return. Once brands can see objectively that their spend is being used well, revenue follows, which is why he attributes a large part of the ads growth to this.
- **Internally.** More output at the same headcount, and a small engineering team building an amount he describes as mind-blowing. He says publicly that they have cut almost all their third-party software spend to zero, along with much of their managed services, and automated manual work that people used to do.

Roughly 500 engineers plus around 150 people in data science, analytics, product and design, and still hiring. He is careful to say they consider themselves early stage rather than a large slow company.

PARTING ADVICE

Asked how he kept upgrading himself fast enough to run a business of that complexity at that age, he refuses the framing and gives credit to the people around him. He notes that after everything they have built, he still measures Zepto against companies elsewhere and feels there is a long way to go.

What he says actually worked: they were able to attract senior leaders with decades of experience who believed in the vision early despite being far more experienced than the founders. Working shoulder to shoulder with an exceptional finance leader or operations leader forced him to pull his boots up and learn.

And the mechanism, which he is refreshingly blunt about: he asks the most basic questions. How does accounting policy work. How does this piece of hardware work. He says they look stupid internally quite often, and that this is the point, because you shamelessly ask, shamelessly learn, aggregate what many people know, and grow. Surround yourself with people smarter than you and learn from them shamelessly.

CHAPTER 3

Varun Vummadi: Turning Down The Safe Offer



“Why Two IIT Engineers Turned Down \$550K Jobs To Build A Startup”

Varun Vummadi Co-Founder, Giga. Interviewed by a YC partner, 29 May 2026.

Video: <https://www.youtube.com/watch?v=2Ap1dnv-GXA>

3.1 What the company does

Giga builds AI agents for customer support, working with very large companies including a major food delivery platform, one of the largest crypto exchanges, and a top-three global telecom provider.

His description of why it matters is concrete. Traditionally you call support, hit an automated menu or a basic chatbot, and eventually reach a person, with only 10 to 15% of cases resolved without one. With their system the experience is human-like and 60 to 70% resolve without a person, with an aim of 90 to 95% for their best customers. The point being that you never sit on hold.

3.2 A small town, then Kaggle, then a rejection that worked

His parents were both government teachers in a small town in Andhra Pradesh, and wanted him to be an engineer or a doctor. He got into IIT Kharagpur for electrical engineering, did little for two years

because of the pandemic, and started research on language models in his third year, before the current wave.

He then received a large offer from a leading quantitative trading firm in New York and a PhD place, and then ChatGPT launched. He and his co-founder, friends since their first year, decided to apply to YC on the reasoning that nothing could go wrong by trying. They were three days from starting their jobs when they got in.

THE ONE THING

The interview went nothing like he had prepared for. He had spoken to many founders who all coached him on the idea and the market. Instead their partner told them the education idea would not work and that they should pick something else, on the basis that they had genuine research experience with these models.

He was, in his own word, panicked, and thought it had gone so badly that they would not get in. He says plainly that the company would not exist without that partner taking a bet on them rather than on the idea.

On his own background as an engineer, he is candid about the contrast with his co-founder, who ranked third across the whole institute and received a single B because of an attendance incident. He describes himself as the opposite: bad grades, not much studying, and a lot of Kaggle competitions, entered mainly because you could win money. He made roughly \$50,000 that way, which is how he landed the trading job, and gamed the system enough that he was eventually banned.

3.3 Two pivots, both driven by customers

The first pivot was away from education, after their partner arranged for them to speak to operators from successful companies in that space, all of whom said it was a bad idea. They pivoted about a month into the batch.

They landed on fine-tuning, having read a research paper about reducing model costs, reasoning that tuning a small model might beat caching a large one. They open-sourced models, topped public benchmarks, got attention, and raised a \$4 million seed.

WATCH OUT

He is unusually direct that fine-tuning was a bad market, and explains why in a way that generalizes. The only reasons to fine-tune were cost and speed. The other use case was security-sensitive customers, and selling to large insurance or healthcare companies is a sales process rather than an engineering one, which did not suit them. It took about a year to realize this.

THE ONE THING

How they found the real business: they looked at which of their customers were growing well, and only two categories were, customer support and coding. So they chose support. He stresses this was observed from customers rather than reasoned from market size, and that their first customer in the category was another company speaking at this same event.

3.3.1 Beating a much larger competitor

By the time they committed to support, there were already well-funded competitors with famous founders. Two honest answers about why that did not stop them: they did not know one of the main ones existed when they signed their first customer, and they did not think about competition much anyway. The mentality was whether customers would pay and whether they could deliver value.

Then the real contest. Eight people against a company with several hundred employees, competing for a large food delivery platform's business, and they won.

THE ONE THING

His conclusion from that: there is a lot of arbitrage available in actually building a great product rather than building a sales team.

Asked why a very large company would trust eight people, he does not pretend it was purely merit. He credits an introduction through YC to a founder of a fellow YC company, which created inherent trust. Then they piloted for three months, never went down, and the metrics were good. He notes the customer was genuinely meritocratic, and that having won them, other large public companies followed.

3.4 What he thinks agent products actually reduce to

The most reusable idea in the talk, and he claims it holds for essentially any company of this kind.

THE ONE THING

The policies are a text file, and the work is iterating on that text file to move a business number. For support that number is resolution rate or satisfaction. You may start at 30 or 40% resolution; everything that matters is how you iteratively get to 90%.

He says the same shape applies to compliance and internal IT support, which is why some of the largest consumer companies are piloting them for internal use rather than customer-facing work.

His current bet follows from it. He argues the real bottleneck in every enterprise deployment is the forward deployed engineer: people who sit with the customer and configure the system. So they are building an AI version of that role, which joins the customer's chat and meetings, takes notes, and makes the configuration changes itself.

3.5 Running the company on automation

One of their stated values is automate, automate, automate, and the overall mission is to automate the world's work, so they push people to use as much of it as possible. Examples he gives run from scheduling to sales, where the team pulls call transcripts to analyze what arguments work against a particular competitor.

THE ONE THING

Asked how many engineers they would need without coding agents, he estimates six to seven times as many. But his reason for valuing it is not cost. It is the absence of context switching: one person owning and building the whole thing ships faster, and transferring context between people kills and slows things down.

TRY THIS

Their interview process is worth stealing or at least arguing with. Candidates write code with the tools available, and then have the tools removed and are asked to change the code. The intent is to check they understand how the code works. He is open that this keeps evolving and openly wonders whether it will still make sense as models improve.

The other thing they screen for is spikiness: some extraordinary ability in the top fraction of a percent. He gives his own and his co-founder's job offers and ranking as the examples they calibrated against.

3.5.1 A mistake he names about himself

Asked whether lacking a business background hurt, he first gives the practical answer: plenty of people will buy from you without one, you just have to find the right buyer, and neither of their first two large customers cared about the size of their sales team.

AGAINST THE GRAIN

Then the admission. He had a significant debate with his co-founder because he believed sales was the most important thing in the company, and he says he was wrong in many ways. His evidence: look at the successful companies in this field and it is product. The major labs are not winning on their sales teams and do not even pay sales commissions.

His revised view: product is the most important thing. How good is it at delivering a lot of value to the customer in a short time. Prove that and the rest follows.

3.6 The decision his family disagreed with

The video is named for turning down the offer, and he does not smooth over what that cost.

His father was, in his words, super mad, and it caused a serious argument at home. Coming from a middle-class family, a high-paying job carried a lot of expectation, and his parents had been happy about it. He tried to explain by showing them what YC was and which companies had come through it, and by pointing out that if it did not work within a year or two he could get a job again.

He also admits the doubt was not only theirs. He wondered internally whether he was doing something wrong.

THE ONE THING

What he and his co-founder actually wanted was to see how far they could go. He describes it as wanting to reach their potential and test whether this was the best they could do, and says the same mentality is why they later declined acquisition offers. He attributes it more to his co-founder, who he describes as fundamentally not motivated by money.

3.6.1 On ideas, and on charging

His self-criticism is that even after getting into YC they worked on several ideas that made no revenue for a long time. Ideas are abundant and you can generate ten of them in a minute; the question is whether anyone will pay.

TRY THIS

What they now do for new products: confirm the customer can actually pay, predict the amount, and get a commitment before building it.

Asked why charging early matters when you are a student nobody will pay, his answer is that if the problem is important enough, people pay, either with money or with time. For any business selling to other businesses, if the problem matters they should be willing to pay money. Otherwise you are solving a fake problem.

On where to be based, he is undogmatic: stay close to customers wherever they are. If your work is research-heavy, he thinks the Bay Area is genuinely the place, because access to researchers is unmatched. If your customers are in India, be in India.

PARTING ADVICE

Just get started, and start trying to sell the thing. He thinks burning the boats has real value, because things only become properly real then. He describes the moment when the company was not working and both of them were thinking about the offers they had rejected, and says that pressure forces you to make things.

He immediately qualifies it, which makes it better advice: it is not really burning the boats. If you have a job, you can get another one. It is not going anywhere.

And the version for right now: the cost of building things is so low that people should build things, try to deliver as much value as they can to very small customers, and see whether those customers will pay them.

CHAPTER 4

Mukund Jha: Living At The Edge



“Emergent: How Six Months of Tinkering Led To A \$100M ARR Company”

Mukund Jha Co-founder, Emergent. Interviewed by a YC partner, 6 June 2026.

Video: <https://www.youtube.com/watch?v=yyXCQH55N4>

4.1 Nine months to \$100 million

Emergent lets someone with no programming knowledge build software they can actually ship, that users can use, and that they can charge for. They started as a research lab building coding agents with a team of four, reached first place on the standard coding benchmark, and then asked what would happen if this were available to everyone.

The numbers, nine months after launching the current product: more than 8.5 million users, over 10 million applications built, and an annualised run rate past \$100 million. Users in 190 countries, with most revenue from the US and Europe and about 10% from India.

THE ONE THING

His framing of why this is worth doing sits underneath everything else. If you removed the software companies from the major stock indices over the last thirty years, the line would be roughly flat. So the question became what happens if you give that power to almost everybody. There are a billion people with ideas, and most of those ideas die for lack of access to a way of

building them.

4.2 Five startups, and a habit of following intuition

Emergent is his fifth. He grew up in a middle-class family, studied engineering, and was inspired early by watching Steve Jobs, to the point that he bought an iPhone as a souvenir on an internship in Spain in 2008 even though it did not work in India.

He went to the US for a PhD in 2009, interned at Google, discovered his intended research had already been done there two years earlier, and dropped out to join. He was the youngest person on a 50-person team controlling search ranking, at a time when the company was resistant to machine learning and he was a machine learning engineer. He credits being young with the license to ask why they were not using it, and ended up pushing some of the larger changes of that period.

Then a group education platform, which pivoted into something he was not passionate about, so he returned the money and shut it down. Then a habit-formation company, which failed partly for a logistical reason he is honest about: he moved to India while his engineering team stayed in New York, and discovered the hard way that coordinating that was very difficult at the time.

THE ONE THING

The thread he draws through all of them is trusting his intuition more, and specifically starting from a personal problem. Dunzo began because he moved to Bangalore and had too many errands, so he started a WhatsApp group and gave the number to friends. Whenever he has solved a problem he personally had, the feedback was stronger, because you relate to the problem and the customer more deeply. Emergent came the same way: he and his co-founder, his twin brother, both have far more ideas than they can build.

4.2.1 What the previous company taught him

Dunzo reached roughly 10 million monthly orders, around a million delivery riders, and some 5,000 stores. He describes the ending as bittersweet and takes three lessons:

- **Solve the hard part.** There were around 87 companies doing the same thing, because the entry point was easy: message us and we act as a concierge. The hard part was the last mile, making sure the customer actually receives the product in the right condition. They chose to do that, including doing deliveries themselves early on, which he connects to the advice about doing things that do not scale.
- **Focus.** The failure he names specifically. Their small-warehouse model was working well while they were simultaneously running a marketplace, a pickup and drop service, and several other things. Recognising what was working and doubling down would have helped a great deal.
- **Operational rigor transfers.** They had a team whose whole job was watching every order, effectively a permanent war room, because operational things break constantly. He has carried that directly across: Emergent monitors everything being built and flags what breaks.

4.3 Six months of tinkering with no objective

The part of the story the video is named for, and he does not dress it up.

He left Dunzo in September 2023 and was, in his words, pretty depressed, not wanting to do anything. For six months he reflected on what they could have done better. The timing was lucky, because the current wave was beginning, and building became his escape from the noise. He would spend ten or twelve hours tinkering with everything that came out: new voice models, new open models. He built himself an assistant on his laptop that he could talk to.

THE ONE THING

He describes it as the luxury of six months of pure tinkering on things he liked with no objective in mind. What came out of it was a conviction that coding as a space would be disrupted very quickly.

His argument for why this works: when you build for the pure joy of it, because you want to solve a problem, you go genuinely deep and surface insights that are otherwise very hard to get. All the insights from that period went into the product.

4.3.1 Being rejected for the ambitious version

When they started, the fashionable thing was to build copilots, and that is what investors wanted to hear. They pitched ten or twelve and were rejected by most, despite him being a founder just out of a large company, because they said they were going to automate software engineering and the investors thought it was crazy and the technology was not there.

AGAINST THE GRAIN

His counter was not that the models could do it, but that you could see where they were heading. They took the view that progress would be exponential and that they would always build in the direction of where it was going, which let them approach the problem as automating the whole of software engineering rather than a piece at a time.

The interviewer gives this a name worth borrowing: living at the edge. The best startup ideas come from things that are not quite possible yet. You could already see the sparks.

TRY THIS

The most concrete version of that discipline. When they started, models could not reliably produce structured output, and he counted at least twenty or thirty companies in their own YC batch solving exactly that problem. They took the view that the next model would fix it, skipped the problem entirely, and built the agent instead.

So the exercise is to work out which of the problems in front of you are about to stop being problems, and refuse to spend your company on those.

4.4 What is actually under the product

They built a multi-agent system at a time when the word was not in use: different agents entering at different points to do different things, including one that tests the application and one that designs it, coordinated through a large memory system that learns.

THE ONE THING

The compounding mechanism is the interesting part. Every time a new application is built on the platform, their agents extract what is learnable from it and store it, so every new application makes the platform better. On top of that data they do reinforcement learning and some fine-tuning.

Because nobody else was working at that layer, they had to invent much of the infrastructure themselves. His example is container technology: they wanted to preserve state so multiple agents could run in parallel from the same snapshot, which meant building disk and memory snapshotting. His point is that living at the edge means you hit these problems before the ecosystem does, and often have to solve them yourself.

WATCH OUT

The cost of that position, stated plainly. When a genuinely new class of model arrives, you have to delete what you have learned and reimagine the product from the perspective of the new model. In nine months they have rewritten their system three times. Each time a new model appears they have to rethink what possibilities open up and where that model will be in six months.

4.4.1 Why they attacked a benchmark

An honest and slightly comic origin story. During the batch they had a new idea every week, the team was getting frustrated, and he needed to give them something stable to work on while he figured out direction. So he picked the hardest coding benchmark available, partly as a distraction.

It took three months and they became first in the world on it. That turned out to be the foundation for everything: parallel test-time compute, the memory system, agent-to-agent communication. All of it was discovered while working on the benchmark.

THE ONE THING

The generalizable lesson he draws: attaching yourself to a number that shows progress is a very good way to attack a goal, because it focuses you in the right direction and gives fast feedback. Note that they did this *before* having a startup idea, and the idea came out of it.

4.5 Not being first, and winning anyway

Emergent was not the first product of its kind, much as Dunzo was not the first of its kind among 87 competitors.

His read on the difference is a customer insight rather than a technical one. Other platforms were mostly focused on the front end and on producing demonstrations, and that is where they were finding traction. But users were going to want real software that ships. Most solutions were good at getting started and bad at finishing: no working back end, no real database, no working result.

THE ONE THING

So they approached it from the opposite end, asking how you would automate all of software engineering, and built nearly everything from the ground up. When they ran the same prompts across every platform, they were substantially outperforming. The insight was that consumers want real working software rather than prototypes, and nobody was solving that.

On distribution, he describes converting growth into an arithmetic problem: how many views do we need, how many impressions, how many clicks, how many users. Knowing the product was strong, the constraint was simply getting in front of people, and working influencers turned out to be the right channel.

The team is roughly 95% in Bangalore with a small and recent presence in San Francisco, and he mentions hiring for learning slope: people genuinely excited about solving these problems. What he thinks separates them is that everyone in the company enjoys working with this technology day to day.

PARTING ADVICE

Three pieces, and the first is the one most specific to this audience.

Think global from day one. Having built both a local company and a global one, his realisation is that they take exactly the same effort. It is equally hard either way. And since the reach, the access and the technology are now the same for everyone, you can serve a global customer from day zero.

Follow your intuition. You will get a lot of advice, and as a founder you probably have a better

sense of what your customer wants and needs.

Then multiply it. Take whatever you are thinking now and make it ten or a hundred times bigger. His phrasing: this is not the time to attack the floor, it is the time to attack the ceiling. He adds the recruiting argument for it, which recurs across this event: starting on a harder idea is in some ways easier, because you can inspire more people to join you on a harder problem.

CHAPTER 5

Vidit Aatrey: Rigid On The Problem, Flexible On The Solution



“How Meesho Became India’s Biggest Shopping App”

Vidit Aatrey CEO, Meesho. Interviewed by a YC partner, 11 June 2026.

Video: https://www.youtube.com/watch?v=49L8lVe_PVo

5.1 The scale, and the gap it is aimed at

Meesho is an e-commerce platform focused on giving the best value for money across categories, built for mass-market India from the start. Their mission has not changed since 2015: bring a billion consumers and every business in India online.

The numbers he gives for the last twelve months: about 250 million people bought something, close to a million sellers, and roughly 2.5 billion orders, with those buyers averaging around ten purchases a year. That 250 million is still growing over 30% year on year.

THE ONE THING

The reason there is room to keep growing is the part worth internalizing. In a country of a billion and a half people, only somewhere between 350 and 400 million buy anything online in a year. So a very large number of people are still not in the ecosystem at all, and that gap is what the

company is pointed at.

5.2 The observation that started it

He and his co-founder both come from small towns. In 2015, two years out of college, they were excited about the mobile internet wave and noticed something that would not go away: in Bangalore everyone around them bought things online, and when they each went home to their families, nobody bought online and nobody sold online.

The conventional view at the time was that e-commerce in India was settled, with the large players established and nothing left to solve. Their belief was simply that something was missing, because none of their own families were participating.

He was working at an ad technology company where he could see budgets from across the startup ecosystem, which is how he saw mobile taking off, and called his co-founder in Tokyo to say the time was now. His co-founder resigned the next day and came back a week later, having already been thinking he wanted something faster moving. They had been close friends through four years of college in the same department and hostel wing.

5.3 Version one, and a mistake he calls stupid

They went to speak to the small businesses that were not selling online, walking into shops across categories and asking why. The answer was consistent: they wanted to sell online but did not know how to find customers online, and the people who bought from them lived within a certain radius and valued their service and curation.

So they built Fashionear, which let a local shop sell to customers nearby. They shut it down three months later.

WATCH OUT

The mistake, in his own words: they spoke only to sellers and never to consumers. When they pushed the app, consumers told them it was the worst of both worlds. In a mall you can touch and try the product; online you get the full selection. This gave them a limited local selection *and* no ability to touch the product.

He calls it the most obvious mistake you can make, understanding one side of a marketplace well and the other not at all. The thing he does feel good about is that they did not spend a year or two on it. They built the app, got hundreds of customers to try it, realized quickly, and moved on.

5.3.1 Version two: observe rather than ask

They went back to the same small businesses, but this time deliberately sat in the shops from morning to evening instead of interviewing. His reason is precise: people articulate problems through their own understanding of them, which may not point at the right solution.

What they found by watching was that shops which said they were not online *were* online. When someone bought something, the shopkeeper took their phone number and added them to a WhatsApp group, then posted new products there. The group was the shop: easy for the shopkeeper to run, low friction for the customer.

So they built a product to make that better, solving the problems around it: collecting payments, and keeping stock updated so orders were not placed for things already sold. That was the product they took to YC.

WATCH OUT

It did not monetize. Small businesses in India were largely unwilling to pay for software, and the response to being charged was that they could manage the way they had before. They were short of money and could not raise their next round.

Worth noting how this looks from outside: hundreds of thousands of shops were using it. That could easily have been read as product-market fit. His judgment was that it could not scale another tenfold, because not enough of those users were active and not enough would pay.

5.4 Finding the power users

The pivot that worked came from asking who was *most* active, and the answer was a completely different profile: businesses with no physical shop at all, selling only through WhatsApp groups. Resellers, in local terms.

The contrast is the insight. An offline shopkeeper used the product for a few days, because having an online shop looked good but updating stock daily was work, and if the online business was the smaller share, why change. The online-native users were in the product all day.

So they went to those users' homes across Bangalore and beyond and asked how they ran their businesses. The biggest constraint was access to suppliers and stock. Many of them were women running a business from home, and they had chosen reselling precisely because it required no money up front: you connect with suppliers who ship directly to the consumer.

THE ONE THING

They built a separate app that gave those users access to supply, listed independently on the store. It grew like nothing they had seen, and he says that is when he understood what product-market fit means. His line: until you see it, you do not know what it is.

What it looked like concretely: they spent nothing on marketing for ten months, because they had no money, and doubled every single month on organic discovery with very high retention. People would complain about missing features and still use the app fifteen or twenty times a day, because it solved something core enough that they invested their own time in improving it.

They then shut down the original app and renamed the new one Meesho.

5.4.1 Why WhatsApp was the right surface at the time

The underlying reason, which he explains as the macro picture, is a good example of noticing a constraint nobody else was pricing in. In 2016 mobile data in India was expensive, and shopping apps are image-heavy. Customers would browse, receive messages telling them money had been deducted for data, and uninstall the app permanently because it felt expensive.

On WhatsApp you choose when to download an image, most of it is text, and images are compressed. So people who would never install a shopping app would buy over WhatsApp. By 2020 they had 10 million WhatsApp groups running in India for the sole purpose of buying from Meesho, meaning 10 million sellers.

5.5 Killing a working business

The hardest decision in the talk, and the clearest illustration of their stated value.

Two things changed at once. The cost of data fell to effectively zero, which removed the problem they had built the whole model around. And the pandemic forced large numbers of people to learn to buy online. They realized that if they did not move consumers onto their own app, they would eventually lose them to someone else.

WATCH OUT

He was clear it could not be run as an experiment, and the reasoning is sharp. The day you go directly to consumers, the resellers who built your business hate you, because you are cutting them out. So a half-commitment loses both: you do not win the consumers and you lose the group owners, and you have nothing left. You have to commit.

Some investors on the board wanted to trial it for a while. They were already a unicorn with significant money raised, so this was not a small conversation.

THE ONE THING

What gave them conviction was the same thing as every time before: they went and spoke to consumers. He, his co-founder, and much of the team went out and found that every assumption they held about those customers was falling away. Nobody minded downloading an app any more. Many of the comprehension problems had gone, though he is careful to say a large part of India still has them.

The timing helped, since capital was abundant and going direct to consumers requires a lot of money to build awareness. And he credits having investors who took the long-term view.

The comparison he draws is the strongest evidence: several other companies had a similar product in 2021. None of them pivoted, and none of them are known any more. Most shut down or were acquired. His summary of why is not strategic, it is emotional: it is hard to kill your existing business and start a new one.

They launched the app in early July 2021 and were the number one shopping app on Android in India within days. As of this talk in 2026, they have held that position every single day since. Their user base went from 10 million to 100 million monthly users in five months.

THE ONE THING

The company value he names for this, which is the most portable idea in the chapter: be problem first. Be very rigid with your problem and very flexible with your solution. The mission has not changed since day one, and the solutions, from local shops to WhatsApp to their own app, keep changing.

And the skill he says a consumer company must have: be the closest to your consumer. Know things about them that none of your competitors and nobody in the ecosystem knows, so you can take contrarian bets nobody else will take. He is careful to add that many such bets do not work out, and theirs happened to.

5.6 The next version, aimed at the same gap

He expects AI to change everything about how e-commerce and logistics run, and describes it as the same moment as 2021. The internal reminder is that all their good decisions focused on the long term while managing the short term, and the question is what you would do if you were starting the company today with the same mission.

THE ONE THING

He frames all eleven years of innovation as two axes, which is a genuinely useful lens: *accessibility*, removing barriers so more people try the product, and *affordability*, helping people do more with less budget. WhatsApp was an accessibility move. Their logistics approach was an affordability move.

His argument for why the next 250 million to a billion probably happens through AI is grounded

in specific observations from rural areas. People cannot use the app: they do not understand what is written even in their own language, do not know what adding to a cart means because there is no offline equivalent, do not understand what a rating or a review is, and are afraid that clicking a button will take money from their account. They are overwhelmed, and they are most of who is left to serve.

So they built a voice product with an explicit design constraint: the customer never has to read anything, type anything, or press a button. Images appear and disappear, you say what you like, you say your address aloud. The software becomes invisible. He notes this is a vision rather than a finished thing, and that they are paranoid about doing it before anyone else, because nobody else should reach a billion people first. He allows that soon they may not have an app at all.

PARTING ADVICE

He thinks this moment is the equivalent of 2015 and 2016 for a new paradigm. Back then mobile internet was taking off and they got excited and jumped in. It is early, everyone is unclear what happens in two or three years, and you can form a point of view and jump in.

If he were starting today he says he would do exactly what he did then: look at what people around him are doing, then go back to his hometown and look at what people are doing there, and find what is not happening. The difference is that the gaps he would find can now be bridged. You can no longer say people are not using something because they do not understand it or are not technical. Everything can be reimagined, especially for a country where people find it overwhelming.

Same method. Very different output.

CHAPTER 6

Lalit Keshre: Love It Or Hate It



"Groww: If Your Customers Don't Love It or Hate It, You've Already Lost"

Lalit Keshre CEO, Groww. Interviewed by a YC partner, 15 June 2026.

Video: <https://www.youtube.com/watch?v=ObBaxL2dFzw>

6.1 Third startup, and the first idea failed too

Groww went through YC in early 2018 with a simple aim: let everyone in India invest. It went public in November of the year before this talk.

He is careful to establish that this is not a first-attempt story. Groww is his second venture-backed company, or third counting a student one, and all the previous ones failed. Even Groww's first idea failed, and it was very different from the product people use now.

He grew up in a small city in central India, where his grandmother wanted him to join the civil service so persistently that she turned it into a blessing she gave him daily. He did not know the word startup existed. Learning about Steve Jobs and Apple rewired how he thought about it, and arriving at IIT Bombay around 1999 and 2000, during the dot-com boom, he found everyone around him doing something of their own, even if it was printing t-shirts and shipping them. Being independent looked fun.

6.1.1 The idea that did not work

In 2016, automated advisory products were popular in the US, so they built one. Their reasoning was sound on the surface: in a country of over a billion people, with hundreds of millions already buying online, there were fewer than 20 million investors. That did not make sense, so they built an adviser to serve them.

It did not work.

THE ONE THING

What the failure produced was a signal rather than a lesson. Customers using the product kept asking the same thing: why this product and not that one? And they would talk about specific products that were not even on the app.

That pointed at selection and transparency, which he connects to what his previous employer stood for: you go there, you see everything, you choose. That reframing became the current product.

6.2 Opening everything up, with no way to make money

The counterintuitive decision. Rather than curating, they showed customers everything with full transparency, seamless onboarding and frictionless payments, and accepted that they might have no revenue for some time.

He describes the hypothesis-building explicitly: customers keep asking why this and not that, and asking about products that are not on the app, so put on the products customers are asking for. With thousands of customers asking about different things, put on all of them. Then connect it to what they already believed about Indian customers, which is that they care about choice and transparency.

They launched a year after starting, expecting that a hundred customers in the first month would make them happy. They got 600.

THE ONE THING

The lesson he draws is the most quotable thing in the interview, and it is about risk rather than growth. In a startup there will always be multiple question marks. If you can reduce it to one question mark, that is what reducing risk means.

Their one remaining question mark was monetization. And he adds a caveat specific to consumer businesses: if there are *no* question marks, he does not think it is really a consumer bet.

6.2.1 Four years with no revenue

The monetization question resolved through the same channel that created the product. Customers on their WhatsApp groups, the vocal ones who talked about them publicly, started asking why they offered commission-bearing funds rather than zero-commission ones. Moving to zero commission removes the revenue.

Their reasoning for doing it anyway: look around the world at consumer products, and if something has near-zero customer acquisition cost, very high retention, very high engagement, very high customer love, and is moving a lot of money, it is hard to find a company in that position that does not eventually make money.

So they took the bet and were a zero-revenue company for four years. What eventually unlocked monetization was, again, customer demand: people asking why they only offered funds and not stocks.

6.3 How they actually talked to customers

The tactics here are unusually specific and worth copying.

TRY THIS

With only twenty people on the app, you cannot get insight from twenty people, so they expanded the surface. They created many WhatsApp groups, and as soon as a customer signed up they were put into a personal group so the founders were reachable and the conversation could continue. They answered questions on public forums. He and his co-founders would go to a cinema before a film started and talk to people there.

THE ONE THING

His warning about how to use what you hear: most of the time you do not build what the customer directly asks for. You read between the lines. No customer said “show me every product with full transparency and make payments frictionless.” You have to work out what the actual obstruction is.

He is clear about what product-market fit felt like and how they identified it: buzz, customers talking about them, organic growth where you are not paying to acquire anyone, and retention and engagement rising day over day. Asked how much of early growth was word of mouth, his answer is almost 100%, and he says most growth is still organic today, with customers arriving via friends, cousins, or in the case of older customers, their children.

6.4 Choosing to be regulated on purpose

A strong early decision, and his reasoning generalizes beyond finance.

There are areas that are unregulated, areas that are regulated, and a grey zone, and the first two of those together are a large share of the space. They chose to operate only in the regulated zone, which means obtaining the license first and then doing business.

AGAINST THE GRAIN

His argument is not about safety or virtue. It is that when you start you should make some very strong choices, because a strong choice removes variables. Once the variables are removed, decision-making and execution become much easier.

6.5 Being the power user of your own product

Asked how they maintain product quality at scale, he names one of their ten commandments: obsess over design.

But the mechanism he actually describes is personal use. He cites the advice that if you are building a consumer product, build for yourself, so that you have at least one customer. He says he personally uses the product around two hours a day, and spends roughly another two hours a day talking to customers who use it.

His reason is that quality assurance teams exist, but the real feeling only comes from being a user yourself and being in direct contact with customers, especially the most intense ones, because they catch things.

THE ONE THING

The test he gives his team on any launch, and the line the video is named for. He should start getting one of two kinds of message: this is awesome, I love it; or this is terrible, I hate it. Both are fine.

What is not fine is indifference. If the response is that they do not care, that is the problem, because it means customers do not care about what you launched.

On competition, his position is deliberately inverted. They are always paranoid about disruption, partly because he is conscious that younger people understand their own generation better than he does, which he offers as one of his reasons for being at the event. But he argues it is counterintuitive: the more you worry about the competition, the further you drift from the customer. The best way to handle competition is not to worry about it and instead understand how customer preferences, trends, and the technology are changing.

6.5.1 On using the tools himself

He connects this directly to the customer argument, which makes it more than enthusiasm. If you are far from the technology and not getting your hands dirty, it is very difficult to imagine what is possible, in the same way that being far from customers makes it difficult to imagine how they think.

Having coded years ago and returned to it, his description of the change is that you no longer worry about the housekeeping, and you start noticing what is now possible, which is what you then push inside the company.

THE ONE THING

His before-and-after for building a product is a clean summary of the shift. Previously: hire three engineers, then a product manager to tell them what to do, then a designer, then operations people if there is operational work, then someone to own the revenue and cost. Ten to fifteen people to build a startup.

Now the one person who worked out what needs to be built can sit down and do the design, the product management, the coding, and write automation for the operational work. What stays the same is understanding the customer's need, and you still have to nail that. What has changed enormously is the how.

6.6 Four co-founders who stayed aligned

The interviewer notes that aligning two founders is hard and they have four. His answer has three parts and is the most directly reusable section of the chapter.

TRY THIS

Write the value system down, in pen. His distinction: the value system is written in pen, and strategy is written in pencil and can change year to year. At the start they wrote a very large document covering what they stand for, what they will do, and what compromises and choices they would make. The first line was that they would always be customer first. If that holds, most decision conflicts over ten or twenty years do not arise.

TRY THIS

Assign clear ownership anyway. In a startup everyone does everything, and he is candid that all four of them have at various points been product managers, engineers, and operations people. One took support calls and did identity verification; the finance co-founder wrote code; they made their own early videos, which he cheerfully describes as crappy. Doing everything is fine. What matters is deciding who is responsible and accountable for each area, and who owns the decision if there is a conflict on something major.

The third part is not a mechanism. You have to enjoy their company. His test: weekends are the only time they do not see each other and they are still messaging constantly, and if on a Monday you still want to go to the office because you are going to people you love working with, through the inevitable ups and downs, that is the ideal co-founder relationship.

6.6.1 What is next

The problem he describes is a nice example of a product having to grow with its users. Their customers are becoming wealthier, both because wealth correlates with age and because he thinks their customers are unusually prudent and so compound faster. Someone who joined at 25 is now 35 with significantly more to manage, and their needs change. If the product does not evolve as customers grow, other people will build the alternatives. Hence wealth management, alongside making the product compelling for the youngest investors who can start at 18.

PARTING ADVICE

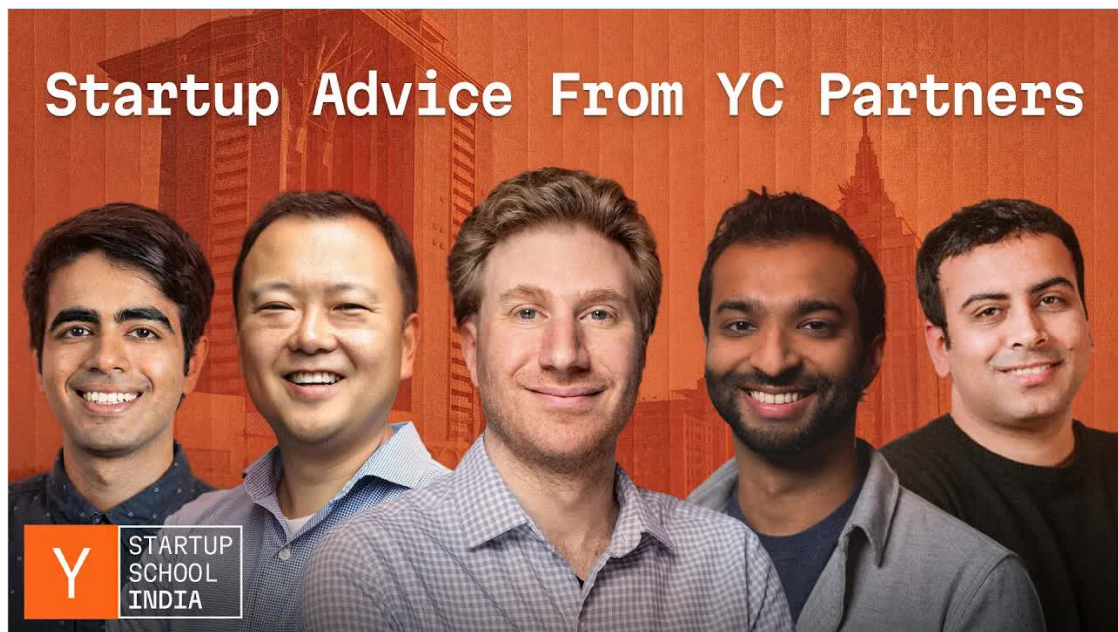
Two pieces, and he gives them in a deliberately self-undermining order.

First: do not listen to older people like him when you do a startup. The world is changing so fast that the advice being given comes from the context of what they saw, and younger people have much better context now.

Then, in spite of that warning, one piece anyway: do something that does not feel like work. Time should blur. He mentions being asked the previous day what sacrifices he had made building Groww, and finding that he could not think of any, because they always enjoyed it. There were low times, but the whole thing was fun, and he thinks that is the most important part.

CHAPTER 7

Closing Panel: India Can Create The Largest AI Companies



“India Can Create The Largest AI Companies”

Closing panel YC partners with two guests, Puneet (formerly SuperDaily, now building again) and Arnav (Peak XV), 27 June 2026.

Video: <https://www.youtube.com/watch?v=Ju8LVdvuxGM>

7.1 A note on this chapter

This was the wrap-up session, pulling together the day’s talks. Two guests introduce themselves by name, so they are named here. The YC partners on stage are not consistently identified in the captions, so where a point comes from one of them it is reported as coming from the panel rather than attributed to an individual.

The two named guests, in their own descriptions: Puneet has been coding since he was ten, built the grocery delivery company SuperDaily to around \$100 million in annual revenue, exited it, worked at YC, was a venture partner, and is now building again. Arnav worked at YC for several years, had a startup that did not fully work out, and is now at the venture firm Peak XV.

7.2 Why this wave is different from the last one

The argument the video is named for, from Puneet, and it turns on one distinction that is worth keeping.

THE ONE THING

The last wave was *local*. Mobile phones tokenized labor: anyone could say they had an hour free and deliver for any of the delivery companies. That produced hyper-local network effects, and so it produced many separate companies in India, Latin America, the US and elsewhere.

This wave is global. And the competition is not about understanding go-to-market or the right business model. It is about whether you are living at the edge of the technology and understand it ten times better than everyone else. His claim is that this is where India is strongest, and that some of the largest companies in the world will come out of it.

He also notes what makes it possible now rather than earlier: over the last ten to fifteen years the ecosystem has learned to design well, build good products, and build technically deep products.

There is a nice arithmetic detail in his introduction. Asked how many engineers SuperDaily had at \$100 million in annual revenue, the answer is effectively two, one of whom was him. Before any of the current tooling existed.

7.2.1 Selling to a market you have never visited

The obvious objection: fine, build globally, but I have never been to the US and that is intimidating.

AGAINST THE GRAIN

His answer is that this used to be genuinely hard. Building a software company meant finding a warm introduction or living in San Francisco for a few years to build the network. What changed is that everyone in the world simultaneously understands this technology matters, so the question is whether you are at the edge of it rather than whether people will take your call.

The proof he offers is deliberately extreme: a company from India in a recent batch cold-emailed US insurance companies and sold to them. Selling to insurers is hard; doing it by cold email as a third-year student in India would previously have been unimaginable.

So the instruction is to discard the assumption that you need warm introductions. People want solutions that produce outcomes, and this is not about whose budget it comes from.

He notes the two founders who spoke earlier that day had never lived in San Francisco before starting their companies.

7.3 On advice, and why the traditional kind may not apply

Arnav's section, and the most carefully argued part of the session, because he takes the opposing case seriously rather than dismissing it.

THE ONE THING

His opening claim: the future of this ecosystem will likely be defined by the people in the room rather than by a prior generation, and the traditional advice of educational systems and of people who are not native to this technology is now largely irrelevant.

The reasoning is about which path is actually risky. The standard advice is to get a high-paying, prestigious, safe job. Some of those jobs may not exist in ten years, or will change dramatically. So the historically safe path might now be the risky one, and the people most insulated may be owners and entrepreneurs.

WATCH OUT

He then argues against himself, which is what makes it credible. It is much harder for an average young person in India to take the risks someone who grew up in Silicon Valley can, given the difference in social safety nets, and he thinks the ecosystem here is booming but not as deep.

So he says plainly: if you come from a humble background and you get a safe, stable, prestigious, high-paying job, you are already in the top 10% of the country and you have done well, and he hopes many people do exactly that. His pitch is aimed only at the people in the room who disagree with the standard advice.

7.3.1 How you develop an independent point of view

Asked directly, he answers from personal experience and admits it is easy to tell someone to be high agency and hard to actually do.

His answer is that you cannot do it without surrounding yourself with people who do it. He thinks that is why certain universities and programs work: founders arrive already very good and leave as far more formidable versions of themselves, and a lot of that is simply people. The fact that a large number of people turned up on a weekend suggests they want to meet others like them.

THE ONE THING

His observation about the specific obstacle: in many educational systems it is not considered cool to be ambitious or to say you want to be. Somewhere like YC it is expected. Being surrounded by that lets you form your own opinion instead of receiving the same advice from people who did it in the past.

And his sharper version: in this field it is dangerous to follow the advice of people who are not native to it, because they are not in the game with you. Choosing to be around people at the edge is a deliberate decision that compounds across a career.

7.4 Why the founders keep getting younger

The panel confirms the average age has fallen across recent batches, and says it is not deliberate.

THE ONE THING

Their explanation: you are no longer limited by your ability to build, you are limited by the pace at which you can learn. For young founders that is a large advantage, because they can build quickly and therefore get insights quickly.

The second thing they see in the best young founders is that they tinker more, following their curiosity and working on things that are barely good enough to be possible. In doing that you beat benchmarks, find where the bottlenecks are, and those bottlenecks are good ideas.

The framing offered for this is to live in the future and find what is missing.

A related empirical point from the panel, drawing on the day's six talks: almost every founder's big idea was not their first idea. A good startup idea is usually not obvious, and you have to be in the field building something to notice it. One panelist describes finding roughly five good startup ideas every time he works on a side project.

7.4.1 Second mover advantage

The panel picks up something striking about the day: most of the ideas that worked were not first to market. They were the third, fourth or forty-second version of the idea to launch.

THE ONE THING

Their explanation is that people with product clarity can now make their ideas real extremely fast, often with more precision than expected. They are dismissive of the idea that these tools inevitably produce low-quality output, arguing that this means you are not using them well.

So the reasonable, if uncomfortable, strategy they describe: find something that is already working, do it better, and beat them. Unless the first entrant has genuine network effects, which few do, you can win on product alone.

7.5 How far you can push the tooling

The most concrete and most quantified section, delivered as a personal account rather than advice.

One panelist describes not realizing how much had changed until a break over the previous Christmas, when he decided to build something end to end and paid for a top-tier usage plan for the first time.

THE ONE THING

His claim: unless you are paying for at least that level of usage, you are not close to the frontier. There is a meaningful unlock from simply letting the usage run.

He is careful about the privilege involved, noting that one person he mentions spends thousands of dollars a day, and that obviously not everyone can. His justification for reporting it anyway is that it lets you see the future, since cost per unit of intelligence should keep falling substantially.

TRY THIS

What actually changes when you stop rationing: do not write the twenty tests a person would write, write ten thousand. Do not write a small amount of documentation, write a lot of it. Do not cover a few edge cases, cover all of them.

He also notes that doing this surfaces startup ideas constantly. Building a side project he made because his existing tool annoyed him produced several new ideas, purely from pushing the models to their limits and noticing that you get a much better result if you stop worrying about the cost.

7.5.1 And for people who cannot afford that

To their credit, this objection is raised from the panel itself rather than left hanging, and gets three answers:

- **Open models are getting genuinely good.** One panelist has been trying several and describes them as cheap and pretty good. His two minds on it: being at the absolute frontier seems very useful for a subset of tasks, especially coding, and for many other tasks not necessarily.
- **The connection back to the day's talks.** He notes that the plan to bring the next billion people online through voice requires the models to work at a price point those people can reach, which implies open models.
- **Or go where the budget is.** Arnav's suggestion: plenty of companies want to give their employees unlimited budgets, so if you are constrained, go and work at one of them and be at the frontier. Being early in the game gives you an advantage in understanding how the tools work.

His addition, which echoes the Emergent chapter: build for the model six months or a year ahead, or at least think through what it will be capable of, and you will be well ahead of the competition on your roadmap.

7.6 What they actually look for

Useful because it corrects specific misconceptions rather than restating principles.

WATCH OUT

The misconception named: that you should write the most impressive or most complicated things in an application. Clarity matters more. If they cannot understand what you are building, it does not work.

On the two qualities they talk about most:

- **Taste** is commonly misread as making products that look good. They frame it as *intention*: are your products and demos intentional, are there insights from customers you have spoken to behind your design choices, and how fast can you get to those insights and build something that expresses them.
- **Agency** is described through an older essay about being relentlessly resourceful. Do you let the conditions of the world happen to you, or do you make things happen?

They add that they are not investing in ideas, that first ideas often are not the ones, and that even ideas which reach millions in revenue after a batch may still be pivoted away from. And they note their reason for caring about these traits is not philosophical: it is that the six founders who spoke that day all had to lead companies through genuinely chaotic periods, and being relentlessly resourceful and obsessed with customers is the consistent thread. One panelist observes that nobody prompted those founders to talk so much about their customers.

TRY THIS

The most actionable thing in the session, with an unusually precise definition. A *project* is when two people build something that was not assigned to them and get someone to use it.

What makes the definition useful is what it excludes. You can complete an entire computer science education and a job successfully and never do one: you did it alone, or it was assigned by a class or an internship, or nobody ever used it, or it stayed a whiteboard exercise. Doing projects, especially young, they argue essentially guarantees you find startup ideas, and it builds exactly the traits described above.

7.6.1 What has not changed

Asked what is different after twenty years in this, the answer is surprisingly little, because the fundamental truths about what makes good founders go back a long way. The panel offers the thought that if you could interview Thomas Edison he would probably sound like the founders from that day: obsessed with customers, with tinkering, and with building at the edge of what the technology allows. If it has not changed since then, it is unlikely to change in the next twenty years.

What has changed is leverage, and who has access to it. That is what allows people to be funded at eighteen and build quickly.

PARTING ADVICE

The closing instructions were practical rather than inspirational, which is a good note to end on.

All six companies that spoke were actively hiring, and the panel points out that one of the better routes to becoming an exceptional founder is working at an exceptional company first.

Then the request they cared most about: do not all go home and lose touch. Projects are best done with two or more people, so stay, meet as many people as possible, exchange numbers, and start something with someone you met there. You might meet your co-founder in the room.

One panelist closes by noting that his own introduction to startups was attending the equivalent event as a college freshman in 2013, where he realized the well-trodden path he was on was not the one he wanted to stay on.