

January 1

Build Something People Want

By Paul Graham (Y Combinator)

If you're not solving a real problem for a real person, you're wasting your time. That's the core of almost every YC rejection and every startup failure I've seen. Don't build something just because it's technically interesting or trendy. Find something painful, ideally something you've personally experienced, and build a better way to solve it.

Here's how you can test this fast: email 10 people you think would use your product. If none of them say, "Can I use this now?", you're either solving the wrong problem or targeting the wrong people. Don't fall in love with your idea, fall in love with the problem. Obsess over whether someone would miss your product if it disappeared tomorrow. If not, pivot now.

January 2

Play Long-Term Games with Long-Term People

By Naval Ravikant (AngelList)

Early on, you need co-founders, first hires, and investors who you'd trust in a foxhole, because that's exactly where you'll be. I don't care how good their resume looks. What matters is their integrity, how they handle stress, and whether you'd want to build with them for 10 years. If you're unsure, walk away.

Write down your principles and expectations on day one. Talk about equity splits, responsibilities, and decision-making upfront, don't postpone the hard conversations. If it feels awkward, good. That means you're doing it right. What ruins most startups isn't lack of code or capital, it's co-founder drama caused by mismatched values and unspoken assumptions.

January 3

Market > Product > Team

By Marc Andreessen (a16z, Netscape)

I know this sounds controversial, but I've backed companies with mediocre products and so-so teams that still crushed it, because they nailed the market timing. The biggest mistake I see founders make is over-optimizing the product without validating if there's a market ready to adopt it.

Here's my advice: before you write a line of code, talk to 50 potential customers. Ask them what they currently use, what's broken, and what they'd pay to fix. If they don't have a painful need, don't build. If they do, get a basic solution in their hands ASAP. A great market can pull an average product to success. But a bad market will kill even the best team with the best product.

January 4

Learn to Pitch Before You're Ready

By Melanie Perkins (Canva)

When we started Canva, we pitched over 100 investors and got turned down constantly. We didn't have a product. We barely had a team. What we had was a deck, a prototype idea, and the audacity to believe we could democratize design. What helped us wasn't perfection, it was the ability to communicate a big, clear vision.

If you're early stage, practice your pitch obsessively. Write it. Rehearse it. Record it. Simplify it. Get feedback. Repeat. You should be able to explain what you're building, why now is the right time, and how it's 10x better than the status quo, in 60 seconds or less. Clarity wins meetings. Momentum wins checks. You'll almost never feel "ready," so get in the room anyway.

January 5

Watch What People Actually Do

By Stewart Butterfield (Slack)

When we launched Slack, we didn't ask people what features they wanted. We watched what they used, what they ignored, and what they hacked. We looked at the usage logs every day. That told us more than any survey ever could. Real user behavior is the most honest feedback you'll ever get.

Build tracking into your product from day one. Don't just count logins, follow workflows. Where do users get stuck? What do power users do that casual ones don't? Kill features that get ignored. Double down on what clicks. Ship fast, learn faster. Your roadmap should be driven by actual usage, not internal brainstorming sessions.

January 6

Do Things That Don't Scale (at First)

By Brian Chesky (Airbnb)

In the early days of Airbnb, Joe and I literally went door to door in New York taking better photos for hosts and helping them write their listings. It didn't scale. It wasn't glamorous. But it showed us what made a great listing, and more importantly, what made a host trust us. That scrappy work was the foundation of our whole product strategy.

Find your early users and obsess over them. Handhold them through onboarding. Fix their bugs personally. Ask them what would make the product magical, and build it that week. You're not too good for manual work, that's how you learn what really matters. Scale comes later. Craft comes first.

January 7

Obsess Over Speed

By Patrick Collison (Stripe)

At Stripe, we always pushed to be the fastest-moving company in our space. Not just fast in engineering, but in decision-making, support, hiring, everything. Speed compounds. If you're twice as fast as the next best startup, that advantage stacks every week. You learn faster, adapt quicker, and ship more.

Speed doesn't mean rushing. It means removing friction. Fewer meetings. Clearer ownership. A bias for action. Most startups die not because they made bad decisions, but because they moved too slow to recover from them. Create a culture where shipping is the default, not the exception. Track how long it takes to go from idea → code → live. Then make that shorter.

January 8

Tell the Ugly Truth Internally

By Ben Horowitz (Andreessen Horowitz)

Founders love optimism. That's fine for external PR, but inside your company, brutal honesty is oxygen. When things go wrong (and they will), your team needs to know the truth fast. Don't sugarcoat bad metrics. Don't hide product flops. If you pretend things are fine, no one will know how to help you fix them.

As a CEO, your job is to create a culture where problems surface early. Do this by leading with vulnerability. Say "I was wrong" often. Encourage postmortems. Reward people for raising issues, not just solving them. If your company gets used to hearing "everything's great" all the time, that's when it's probably dying.