

Early Startup days

- Talking to Customers: The YC Way

Talking to Customers: The YC Way

A short guide to how Y Combinator advises founders to talk to early customers.

1. Talk to users constantly, not occasionally

YC's core mantra is "talk to your users." This isn't a one-time research phase before you build — it's a continuous habit. In the earliest days, founders are encouraged to spend as much time talking to customers as building the product. Paul Graham's advice to "do things that don't scale" often starts here: personally recruiting and interviewing your first users instead of outsourcing it to surveys or a support inbox.

2. Talk to a few users deeply, not many users shallowly

Rather than blasting out surveys, YC pushes founders toward **small numbers of deep, in-person (or video) conversations** — often with 5-10 users repeatedly, rather than 100 users once. The goal is to understand one person's problem so well you could describe it better than they can.

3. Watch what they do, don't just listen to what they say

People are unreliable narrators of their own behavior. YC founders are taught to:

- Ask about specific past behavior ("Tell me about the last time you tried to solve this") instead of hypotheticals ("Would you use a tool that...").

- Watch users actually use the product (screen-share, sit next to them) rather than trust self-reports.
- Treat compliments and "I'd definitely use that" as close to meaningless — the only real signal is action (payment, usage, referral).

This overlaps heavily with the ideas in *The Mom Test*, a book YC frequently recommends: ask questions your own mother couldn't lie to you about, even if she wanted to spare your feelings.

4. Find the problem before you fall in love with the solution

Early conversations should be problem-focused, not pitch-focused. A common YC failure mode is founders who "sell" their idea in every conversation and only hear validation back. Instead:

- Lead with questions about the user's current workflow, pain, and workarounds.
- Resist pitching your solution until you deeply understand the problem.
- Look for people who have hacked together a janky manual solution already — that's a sign of real pain.

5. Find your "10 users who love you" before scaling

A famous YC line (Paul Graham): it's better to have **10 users who love you** than 1,000 who sort of like you. In the earliest days, founders are told to obsess over a tiny group of engaged users, talk to them constantly, and fix things for them personally — even manually, even if it doesn't scale.

6. Do things that don't scale, especially onboarding

YC encourages founders to personally onboard, support, and even use the product *for* early customers by hand. This isn't just customer service — it's a research method. Manually inserting yourself into the customer's workflow surfaces problems no interview would reveal.

7. Close the loop fast

Ship small changes based on what you hear, then go back to the same users and see if it actually solved their problem. The cycle is: **talk** → **build** → **talk again**, on a tight loop, rather than long research phases followed by long build phases.

8. Look for signal in numbers, not just words

Talking to customers isn't just qualitative. YC pushes founders to pair conversations with real usage data — retention, repeat usage, willingness to pay — since what people say in an interview and what they do with their wallet or their time often diverge.

Quick Reference Checklist

- ☐ Talk to users yourself — don't delegate it early on
- ☐ Ask about past behavior, not hypothetical future behavior
- ☐ Watch usage, don't just collect opinions
- ☐ Avoid pitching; focus on understanding the problem first
- ☐ Find and obsess over a small group of true fans
- ☐ Do unscalable things to serve early users personally
- ☐ Loop: talk → build → talk again
- ☐ Validate what you hear against what people actually do

Sources of these ideas: Y Combinator's Startup School and essays/talks by Paul Graham, Michael Seibel, and other YC partners, along with The Mom Test by Rob Fitzpatrick, which YC frequently recommends.