Ravi Mehta on customer research: the gap between what customers say and what they do

PM Daily Digest

2026-11-06

Ravi Mehta on customer research: the gap between what customers say and what they do

By PM Daily Digest • November 6, 2026

Ravi Mehta on avoiding misleading customer research, practical ways to set accuracy expectations for LLM features, and Aakash Gupta's Adobe case for what still protects a product when features are cheap to copy.

Customers' words and behavior diverge

Ravi Mehta's new essay opens with a focus group. When asked which app they'd use the next time they were single, nobody named Tinder, which was the largest dating app by far. Yet nearly everyone had met someone on it in the past year [1]. One question asked people to imagine their future selves. The other asked what they had actually done [1]. The real need was a better way to meet people. That is the need Hinge's "designed to be deleted" pitch speaks to [1].

His working definition of Voice of the Customer covers feedback "in all its forms," including support tickets, sales calls, app reviews and casual conversations. It aims at the job behind a feature request, not at customers designing your product. Research should change priorities and investment, not end up as a forgotten 200-slide readout [1].

On AI, he points two ways: - **Faster loops.** One client runs 60-minute calls: 15 minutes exploring needs, a prototype built while the call continues, and the customer trying it in the last 15. That's a loop that used to take a six-week sprint [1]. Interviews also build up a knowledge base that later questions can draw on, so a new question doesn't always need a new study [1]. - **More distance.** AI "can also put one more layer between us" and customers [1]. PMs should check AI-generated conclusions against the underlying evidence [1].

The responsibility changes with level. Directors build reliable research practices. VPs and CPOs settle trade-offs between short-term results and long-term customer value [1].

When stakeholders expect LLM features to be 100% right

The pattern on r/ProductManagement: an owner sees a good demo, and then the first wrong answer becomes "the AI does not work" [2]. The original poster's tactics: - Agree an error budget in the stakeholder's own words, such as how many of 100 invoices a person can fix by hand. - Measure the current manual process's error rate as the baseline. - Sort errors by cost. - Add cheap human review where mistakes are expensive. - Show failures yourself before stakeholders find them [2].

What doesn't help, the poster says, is explaining how the model works [2].

The most practical reply suggests a pilot on a slice of real work. A person approves every output, and the team tracks the rate of outputs accepted unchanged. Agree in advance on the threshold for switching it on and the kind of mistake that shuts it off. That team brought "92% as-is over three weeks" to senior stakeholders [3]. Another commenter keeps a repeatedly charted eval set on the tracking page, since accuracy varies between runs and between models [4].

What protects a product when features are cheap

Aakash Gupta uses Adobe as the case. It had no down revenue year from 2015 to 2025, and ARR is now $27.5B, yet its market cap fell from $311B to about $96B [5]. His argument: features are "a few prompts" away, and agents weaken muscle memory as a moat [5]. What can't be copied is learning. Firefly has generated 24B+ assets, which he calls 24 billion examples of what people ask for, and Adobe's AI-first ARR grew 150%+ last quarter [5]. His test: write a one-page spec from your own public docs and give it to a coding agent. Whatever it rebuilds is table stakes [5].

An r/startups thread on the same question pushed back on easy answers. Stringing together public APIs isn't a moat [6]. A funded entrant can catch up on 1.5 years of work once you've validated the market [7].

AI adoption is a team problem

Hiten Shah argues that most companies turned AI adoption into homework. A few employees became power users, but "almost none of that learning changed how the teams worked together." His conclusion: "The team has to be the unit of AI adoption" [8]. Figma described its version as a "messy middle." Domain specialists build prototypes, and Figma wants PM-like functions in its CIO org to find what's working and build shared foundations for others to build on [9].

Figma also says its embedded data scientists build as much or more product intuition than its PMs and engineers [9].

Also

- **Agent UX.** Paweł Huryn's Mini Harness is about 1,200 lines of Python. In his 10-task, five-run test, he places it in the top group with Claude Code but doesn't call it the winner [10]. His PM lesson is that error messages are instructions to the agent, and "the words the agent reads are part of the product" [10].
- **Agentic commerce.** Stripe is hiring engineers and product people for agentic-commerce problems such as wallets, identity, risk, discovery and micropayments [11]. Patrick Collison says "pretty much every part of the stack is going to change" [12].

Sources

1. Voice of the Customer.
2. r/ProductManagement post by u/aureliatech
3. r/ProductManagement comment by u/TheInitialAgency
4. r/ProductManagement comment by u/xasdfxx
5. substack
6. r/startups comment by u/NiagaraThistle
7. r/startups comment by u/Just_Look_Around_You
8. X post by @hnshah
9. How AI Is Turning Figma's Finance Team Into Builders
10. Agent Harness: How I Built My Own Claude Code, Step by Step
11. X post by @jeff_weinstein
12. X post by @patrickc