

Lenny's moat test vs. Paul Graham's "just grow," plus what builders of AI skills are learning

PM Daily Digest

2026-10-14

Lenny's moat test vs. Paul Graham's "just grow," plus what builders of AI skills are learning

By PM Daily Digest • October 14, 2026

Two views on startup defensibility came out on the same day. Also: Hiten Shah on testing AI skills, Sachin Rekhi's AI-fluency rubric, Teresa Torres on engineers joining discovery, and PM frustrations on Reddit.

Moats: a test for them, and advice not to worry about them

Lenny's Newsletter published a three-step test for whether a startup has a moat. First, identify a compounding advantage. Second, judge how strong it could become. Third, measure whether it is working [1]. Only four advantages count: switching costs, network effects, scale economies and data effects. If you can't see the potential for any of them, "that's a red flag" [1]. Speed, distribution and brand are treated as "fuel," not moats. Distribution matters more than ever because building has become easier while reaching customers hasn't, but it only "helps create the raw ingredients" for a moat [1]. The test is to judge a company by what it actually does, not by its stated strategy: what the team works on every day, what is on the roadmap, and which teams are large and influential [1].

That same evening, Paul Graham said nearly all standard startup advice still applies in the age of AI [2]. On moats specifically: "don't stress about moats... Just grow." In his view, passing on growth because it lacks a moat is far worse than not having one, and "moats often emerge later anyway" [3]. He added that AI can do much of a cofounder's work, but a cofounder's main job is helping bear the stress [4]. If you're evaluating a company to join, use Lenny's test. If you're deciding whether to chase a growth opportunity, Graham's warning applies.

What building expert AI skills teaches

The previous brief covered Hiten Shah’s Linear-vs-Jira example. His full article adds lessons from testing. A feed held several observations of one ClickUp pricing change. Counted separately, they could suggest a competitor kept changing prices. Experts check where information came from, because “plenty of retrieved material can still amount to one source” [5]. Live use exposed failures the evaluations missed. One skill asked users for information the system already had. Another created a file the user couldn’t get to. Shah’s fix was to follow each request all the way to a usable result [5]. Expert corrections that keep recurring can become reusable rules or tests, but test them on several cases, because “a rule that fixes one example might make another worse” [5].

Sachin Rekhi now scores every employee on seven AI-maturity levels: Abstain, Search, Assist, Judge, Automate, Compound, Transform. Assist means producing real deliverables. Judge means critiquing and iterating on output, which he calls critical “in an age of AI slop” [6]. He named each level with a verb so behavior is easy to observe [6].

Teresa Torres on engineers, discovery and training

Torres says the common mistake is that engineers first see the work at sprint planning, after requirements and design are done. That is the most expensive and disruptive point to get their feedback [7]. She wants at least one engineer involved while the team learns from customers and explores solutions [7]. On AI blurring roles, she treats one- or two-person teams using coding agents as “an interim step,” not the end state. PMs who submit PRs need to learn about scalability and maintainability. Engineers who build without customer context run into the same wall from the other side [7].

In a separate episode with Petra Wille, Torres argues that a one-hour talk can’t create a habit-driven team. Know-how transfers fast, but skills need practice over time [8, 9]. Talks work best as part of a longer coaching program [10].

When agents become the users

- **Agent shopping.** Shah suggests that Shopify sellers ask an agent using Tobi’s new Shop skill to shop for their product category without naming their brand, then see whether it recommends them [11].
- **MCPs over UIs.** One B2B team says it does nearly all its work through Claude and uses outside software through MCPs. Its time in software UIs has “decreased by a great margin,” and it expects customers to follow [12].
- **Google Labs hiring.** Josh Woodward, head of Google Labs, the Gemini app and AI Studio, says he is hiring for “unlearning” rate and explosive endurance [13].

From Reddit

In one org, weak product leadership has turned PM into project management. UX, engineering and BAs fill the requirements gaps [14]. Suggested fixes: track the cost in clarification hours, rework and bounced tickets [15], and don't start work without a problem statement, success metrics and acceptance criteria [16]. A thread on PM frustrations in 2026 offered a prioritization tactic. Rank the backlog against named strategic goals, make each new request say what it would displace, and record who overrode the order so you can review those calls at quarter's end [17]. One commenter described an executive who compared a two-week vibe-coded build with a six-month estimate, even though the quick version broke legal and contractual obligations [18].

Sources

1. How to tell if a startup has a moat
2. X post by @paulg
3. X post by @paulg
4. X post by @paulg
5. X article by @hnshah
6. X post by @sachinrekhi
7. How Engineers Can Get Involved in Product Work | Teresa Torres (Continuous Discovery Habits author)
8. X post by @ttorres
9. X post by @ttorres
10. X post by @ttorres
11. X post by @hnshah
12. r/ProductManagement comment by u/Mobile_Spot3178
13. X post by @lennysan
14. r/ProductManagement post by u/Evening_Abies_6867
15. r/ProductManagement comment by u/ykwimykwm
16. r/ProductManagement comment by u/CryDue3918
17. r/ProductManagement comment by u/Own_Try4793
18. r/ProductManagement comment by u/danielleiellle