

Vibe-coded clones are cheap now. PMs say the moat is the hard last 10%.

PM Daily Digest

2026-10-19

Vibe-coded clones are cheap now. PMs say the moat is the hard last 10%.

By PM Daily Digest • October 19, 2026

Practitioners on how to defend against AI-built copycats, how to get usable PRDs and briefings out of AI, and why reliability thresholds and craft matter more than how fast you ship.

How hard is it to clone your product?

A PM at a long-established analytics software company asked how to deal with vibe-coded clones. The replies agreed that the first version is cheap now. One commenter said they rebuilt about 70% of a 15+-year-old technical SaaS product in five hours with an LLM, “even down to reproducing the same inputs/outputs” [1]. Others argued that cost is in the remaining work. A demo might take a week and 90% might take a few months, but “the extra 10% will be far more difficult” and ends up needing dedicated hires just to maintain [2]. Another listed what a copycat won’t have: the backlog of fixes, customer success calls, PRDs, corner cases and the backend [3].

One commenter said speed changes how teams have to build. They argued “Agile isn’t fast enough for Agentic development.” In their view, QA harnesses and automated guardrails, not paperwork, are what make AI output trustworthy at volumes nobody can review by hand [4].

Hiten Shah raised a related risk: you can now build software that “looks ready for a million users before you’ve learned what one person actually needs.” His advice is to give early access to people already doing the work: “The faster you can build, the sooner you need contact with reality” [5]. Ravi Mehta makes a similar point from the craft side. AI lets teams afford the “nice-to-haves” they used to cut, yet many are spending that capacity on “an industrial-scale slop

factory” of extra features. He expects the advantage to go to teams that build with “craft, taste, and ambition” before everyone has the same tools [6].

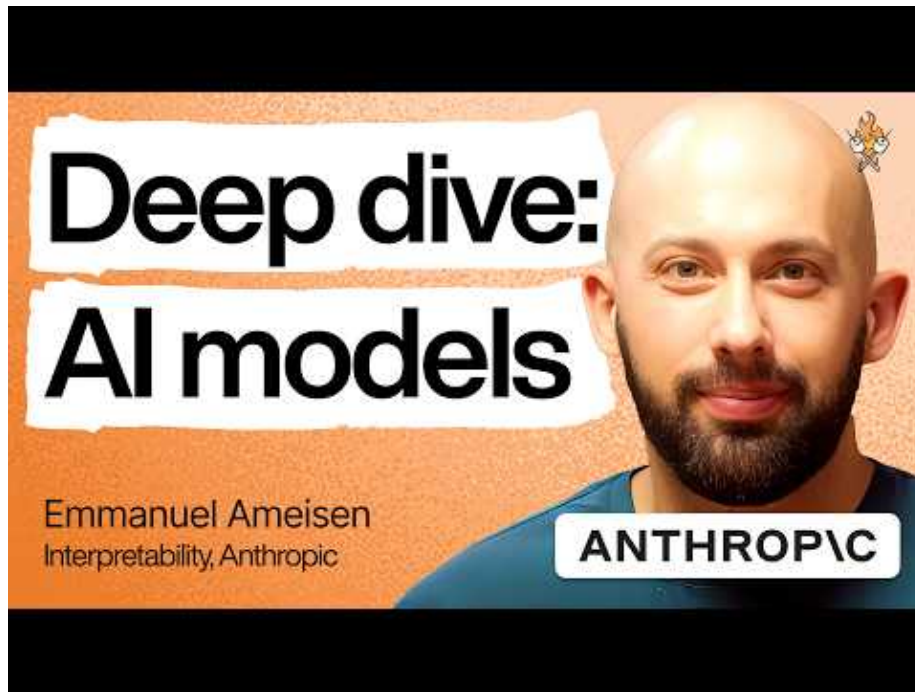
What makes AI output usable

When a PM asked whether AI-written PRDs are acceptable, the answers mostly came down to how much context and checking goes in. One PM said early drafts were weak. After nearly a year of connecting Claude to Jira, using templates and keeping a “second brain,” it “nails it first time pretty much every time” [7]. Another described a full pipeline: 48 rule checks, six quality dimensions, accept/warn/reject gates and LLM revision loops, drawing on 34 product artifacts. They rate the results “9/10 good,” with “the odd one... so wrong it hurts” [8]. A third commenter asked what PRDs are for if developers just feed them to coding agents. Maybe agent-to-agent handoffs need a different format [9].

Productify describes the same pattern for recurring work. A daily Jira/Confluence briefing went through four or five rounds of prompt rewriting. It learned to group tickets by business topic, put customer, payment, revenue, security and compliance risks first, and say what it could not verify [10]. The most important rule was “Do not invent facts,” because the agent would label a single ticket “highly critical” without knowing the wider context [10]. The author’s “Judgment Loop” is: you define good, AI drafts the prompt, you review, and you automate only once you trust it [10]. Write your context down once to stop paying the “Prompt Tax” [10].

Reliability thresholds, not smooth gains

On Lenny’s Podcast, Anthropic interpretability researcher Emmanuel Ameisen explained why AI features seem to suddenly start working. Capability improves smoothly, but each task has its own reliability threshold. An email helper you have to fix half the time wastes your time. One you fix 5–10% of the time is useful. Claude took over most of his coding when it went from roughly 80% correct (“not useful”) to 99% [11]. For AI features, that means judging task-level reliability and how much correcting users have to do.



A mind-expanding conversation on how models actually work / Emmanuel Ameisen (Anthropic) (30:54)

Careers and distribution

- **“Product builder.”** Aakash Gupta notes that LinkedIn replaced its APM program with an Associate Product Builder program. He cites an analysis of 10,000+ PM job descriptions: over 30% call for AI skills, and the top skill is choosing the right problem for AI, ranked ahead of RAG and agents [12]. Separately, Gupta says his “Company OS” answers team questions in Slack with sources and runs shared skills in 21–85 seconds for \$0.08–\$0.32 each [13].
- **Job market.** A mid-career PM asked why landing interviews is harder than four years ago. Commenters pointed to laid-off ex-FAANG candidates, hundreds of AI-tailored applications per posting, openings that go unfilled or get downgraded, and \$45/hour caps on senior technical PM contracts [14]. Large employers in healthcare and payments can find candidates with similar skills plus domain experience [15].
- **Discoverability.** Paul Graham: content on a paywalled site that AIs can’t find effectively “doesn’t exist.” He adds that this isn’t fully true yet but “will only get truer” [16].

Sources

1. r/ProductManagement comment by u/trenhard
2. r/ProductManagement comment by u/Angryguy97
3. r/ProductManagement comment by u/mazzicc
4. r/ProductManagement comment by u/DuskLab
5. X post by @hnshah
6. How AI Helps Product Teams Build Better, Not Just Faster
7. r/ProductManagement comment by u/UnderstandingLow3162
8. r/ProductManagement comment by u/Bernhard-Welzel
9. r/ProductManagement comment by u/NotARussianBot-Real
10. Don't Prompt AI. Delegate to It.
11. A mind-expanding conversation on how models actually work | Emmanuel Ameisen (Anthropic)
12. substack
13. substack
14. r/ProductManagement comment by u/msondo
15. r/ProductManagement comment by u/goddamn2fa
16. X post by @paulg