

Anthropic's 17:1 engineer-to-PM ratio sharpens the debate over what PMs are for

PM Daily Digest

2026-10-22

Anthropic's 17:1 engineer-to-PM ratio sharpens the debate over what PMs are for

By PM Daily Digest • October 22, 2026

Anthropic's internal report on AI and work fueled a wide practitioner argument over whether PMs should write code. Alongside it: warnings about counting output instead of outcomes, and a guide to pricing AI products when both value and cost are lopsided.

The PM role is being redrawn, unevenly

A Mind the Product episode went through Anthropic's internal study of how agents changed its work. Anthropic now has about 17 engineers per PM, and supervising and coordinating roles on technical teams “increasingly involved building and shipping code” [1]. Anthropic's interview scorecards mention a candidate's AI-tool track record 3x as often for technical roles and 4x as often for non-technical roles in 2026 as in 2024. In technical interviews, the biggest increase was in comments about autonomy and judgment, while domain expertise took up a smaller share [1]. The junior share of technical hires dropped from 27% to 9% [1]. There is a caveat: Anthropic sells the tool it studied, and the engineers who adopted agents were already improving fast before they did [1]. The host's takeaway for PMs is that coordination is the shrinking part of the job. What's left is deciding what's worth building, what “done” means, and writing specs clear enough for an agent to run on for hours [1].

A large r/ProductManagement thread showed how that is playing out. The original poster sees job openings that ask them to own the whole cycle, from product decisions to writing the code, and wonders if it's “a way for companies to get one person to do two jobs” [2]. A commenter at a 200-person product/engineering org described the opposite approach. Engineers now run delivery and make more day-to-day scope tradeoffs, which frees PMs for discovery, go-to-market

and impact monitoring [3]. Another team tested letting PMs make small code changes. It found that handling the side effects cost Engineering and QA more than making the changes themselves, so they stopped [4].

More output, same blind spots

Several sources argued that cheap building makes judgment more important, not less. Jeff Patton says teams ship features without knowing whether anyone saw, tried or kept using them, and “the AI is not going to instrument that new feature... unless you ask it to” [5]. Aakash Gupta suggests going from an idea straight to five prototypes, then writing the PRD for the winner as “a decision record more than a permission document.” For AI features, that PRD should include a behavior contract of 15–25 example responses ranging from bad to amazing [6]. Separately, he argues that a strategy’s job is to help the team say no [7]. MrBeast’s CPTO, Shiva Rajaraman, sees the PM question shifting from rationing engineering capacity to “should we really be exploring 10?” He uses the extra capacity to build user-based alternatives next to the founder’s own preference [8].

The other side of this showed up in a popular thread about a Head of Product who spends his days building internal tools and “automations” with Claude “that nobody wants” [9].

Pricing AI when value and cost both spike

In a SaaStr talk on how Clay, Figma and Higgsfield price, the speaker said that in AI products, about 50% of users often consume almost nothing, while 1% drive 50–75% of AI cost. The worst customer he saw burned 380x their revenue in tokens [10]. His point is that a dollar of AI cost does not automatically become a dollar customers will pay [10]. When value is spiky, price on outcomes. When cost is spiky, use fair-use guardrails [10]. His throttling method works in steps: - Group users by type and set a “kill point” where usage gets shut off. - Start a slowdown at that kill point, then move it earlier so it reaches more users. - At each step, watch whether users keep going, cut back, churn or upgrade [10].

From features to responsibilities

Hiten Shah argues that “the smallest useful unit of software may be shifting from a feature to a responsibility”: customers pay for confidence that a job will be handled even when nobody is using the product [11]. He gives Linear’s Loops, introduced in July, as an early step. Teams can set Linear Agent to run recurring or event-triggered work and inspect each run, but in his view that is automation and doesn’t yet take responsibility for the result [11]. He cites a 2026 study finding that 68% of production agents ran ten steps or fewer before a human stepped in [11].

Also notable

- **Legora** raised a \$550M Series E at an \$8.8B valuation, betting that lawyers are moving “from AI assistants to agents that do multi-step work.” It says 140,000 legal professionals use its products [12].
 - **Emet**, launched with Deel, says the best AI model fully solves only 49% of real payroll tasks. It is publishing benchmarks written by in-country experts and scores every model against them, including its own [13].
-

Sources

1. Anthropic Reveals How Its AI agents Work 6× more Than Human
2. r/ProductManagement post by u/Clean_Candidate9817
3. r/ProductManagement comment by u/louizik
4. r/ProductManagement comment by u/NoahtheRed
5. Jeff Patton: AI Doesn’t Change Product Management Fundamentals
6. substack
7. substack
8. MrBeast CPTO on Using AI to Run Every Video Like a Product Launch
9. r/ProductManagement post by u/TheOnlyBlueprint
10. How Clay, Figma, and Higgsfield Price AI, and How to (Finally) Get Your Pricing Right
11. X article by @hnshah
12. X post by @MaxJunestrand
13. X post by @Bouazizalex