

Ravi Mehta: PM-to-engineer ratios measure the organization, not the PM

PM Daily Digest

2026-10-02

Ravi Mehta: PM-to-engineer ratios measure the organization, not the PM

By PM Daily Digest • October 2, 2026

Ravi Mehta argues that cutting PMs doesn't remove the product work. A widely shared [r/ProductManagement](#) thread shows what happens when nobody does that work. Also covered: Vistaly's AI rebuild of its opportunity solution tree tool, and new data on what AI subscriptions are worth at API prices.

Fewer PMs doesn't mean less product work

Ravi Mehta argues that a PM-to-engineer ratio “does not measure the PM's talent, bandwidth, or importance. It measures the organization around the PM” [1]. The same 1:30 ratio can mean a founder with a clear strategy and designers close to customers. It can also mean “an exhausted ticket writer” building whatever sales or the loudest customer asks for [1]. When companies treat PMs as overhead and cut them, the job shrinks to execution. Then nobody talks to customers, owns the vision, connects features to outcomes, or protects the quality bar [1].

His view is that AI “shifts the bottleneck from building products to judging what deserves to ship.” So PMs should spend more time deciding direction and probably less time building [1]. He offers a diagnostic built on his 12 product competencies. For each one, ask who is accountable, who contributes, what evidence shows the work is happening, and what would close the gap [1]. The goal is not a target ratio: “You can eliminate the PM role... But the work remains” [1].

Aakash Gupta's examples of the “full-stack builder” model point the other way. LinkedIn replaced its APM program with an Associate Product Builder program. Rippling's CPO moved planning decks to markdown in a git repo. Some Freshworks teams went from 1 PM per 20 engineers to 1:1 [2].

Practitioners describe the cost when nobody picks up that work. A widely shared `r/ProductManagement` thread describes “slop bombs” from adjacent teams. Leadership is pleased by the higher output and asks, “Why do you need to spend time on discovery” [3]. One commenter blames workload more than the tools: PMs are “asked to do multiple PMs worth of work” with no room to think [4]. Another says people stopped reading docs because they can’t trust them. A one-pager where every claim has a source or is marked as a guess still gets read [5]. Lenny Rachitsky shared a related line from Marty Cagan: “I did not appreciate the lengths that people would go to in order to avoid thinking” [6].

Vistaly’s AI rebuild of opportunity solution trees

On Just Now Possible, Teresa Torres and Vistaly’s founders walk through V2. It is a ground-up rewrite: users upload interviews, get a snapshot of each, and an agentic workflow drafts and updates the opportunity solution tree [7]. Nearly all of V1’s functionality was rebuilt in two and a half months, after three years of building V1, and V1 signups were shut off to protect the rewrite [7]. Lessons for AI PMs:

- **Errors compound.** If an interview snapshot is wrong, every layer above it inherits the error [7].
- **Prompt changes run out.** Balancing two opposing error modes took a repair loop in the orchestration, not a better prompt. A cheap code check (a node with too many children) screens cases before the costlier LLM judge, and the eval became a production guardrail [7].
- **Answer first, then correct.** Users want the answer and the ability to fix it, not step-by-step collaboration. The hardest problem is helping them understand what changed [7].
- **Model upgrades aren’t drop-in.** Prompts are specific to each model and version [7].

On a similar note, a practitioner told someone switching into AI PM to spend three weeks on evaluation rather than vocabulary. Their advice: decide how you’d judge an agent’s output as acceptable, wrong, or needing a human [8].

Building product sense

Deb Liu argues that product sense is learned through repeated feedback, “a comment on a document or feedback on a slide” [9]. Her warning is that AI makes it easy to skip that loop and never learn from mistakes [9]. Her advice: draft the spec yourself, ask “What is one thing I could have done differently?”, and get your reps [9].

Model and subscription costs

The Product Compass tested models on 105 real bugs. Opus 5.5 nearly matched Fable 5.1 (41.7 vs 43) at two-thirds of the cost [10]. GPT-6.1 Sol was slightly stronger and over 10x cheaper than GPT-5.6 Sol on complex tasks, mostly because it needed fewer turns [10]. Separately, the author used up part of several plans' weekly allowances and priced each model call at public API list prices. The chart puts SuperGrok at 190x its price, Claude Max 20x at 45.3x, and ChatGPT's \$100 and \$200 plans at 10.25x [10].

Lenny called a DevDay announcement the “sleeper hit” [11]: ChatGPT subscribers can now use their included usage in more than 16 partner products, including Devin and Notion, by signing in with ChatGPT [12].

Quick hits

- **GitHub for AI PM candidates.** Gupta says AI PM hiring managers check a linked GitHub. His rule: build one small thing a week [13].
- **“Average Intelligence.”** Elena Verna: having an average marketer, engineer, analyst, and designer on hand is “better than some of the teams I’ve had” [14].
- **Reusable research.** Hiten Shah demos a product on Oct 2 that carries findings across tasks, so a competitor’s packaging change found in a pricing review doesn’t have to be re-explained for a launch [15].

Sources

1. Product management is not optional
2. substack
3. r/ProductManagement post by u/socialvegetarian
4. r/ProductManagement comment by u/almaghest
5. r/ProductManagement comment by u/BowDigby
6. X post by @lennysan
7. X post by @ttores
8. r/ProductMgmt comment by u/Parking-Stress-3041
9. Chicken Sexing: How to Build Expertise One Tap at a Time
10. Stop Overpaying for Your AI Subscription: GPT-6.1 Sol, Sonnet 5.5, Opus 5.5
11. X post by @lennysan
12. X post by @thsottiaux
13. substack
14. X post by @lennysan
15. X post by @hnshah