

AI is splitting the PM job apart, and junior PMs may lose the practice that builds judgment

PM Daily Digest

2026-10-06

AI is splitting the PM job apart, and junior PMs may lose the practice that builds judgment

By PM Daily Digest • October 6, 2026

Hiten Shah argues AI products should be built around specific responsibilities, not copies of job titles. Meanwhile agents are moving into group chats, and Andrew Chen asks where their network effects will end up.

The PM job, unbundled

Hiten Shah argues that AI products shouldn't copy a job description whole. A company might want feedback synthesized continuously without that same system owning roadmap prioritization and launch planning. Those duties can come apart once no single person has to carry all the context [1]. Shared context can also pull work together. When Shah's team designed market-analysis software, it chose one responsibility, "keep the company current on its market," instead of copying any single department's view [1].

His sharper point is about who learns the work. Repetition is how exposure turns into judgment: after enough customer calls, you notice the sentence that doesn't fit the pattern. When AI absorbs that repetition, "the company gets the answer faster. The junior person gets fewer reps" [1]. He wants ways for people to inspect inputs, make their own calls, compare them with the system's, and review failures [1]. Put concretely: AI can summarize 100 calls in minutes, "now where does the junior PM get those 100 reps?" [2]

Hiring is changing too. In one reported interview, a pharma GPM asked how a candidate would decide whether to build an internal help chatbot, where the answer can now be "just build the prototype." A senior PM was asked how to plan roadmaps that include AI and data engineers [3]. One commenter's view of what to probe is judgment, not prototyping speed: whose problem it is, what

evidence would change the decision, the smallest safe test, and “what decision did that learning change?” [4]

Agents go social

Instinct now lets early-access users add an agent to group chats, for planning trips, splitting ticket costs, or running carpools. Friends don’t need Instinct to take part [5]. The permission design is worth studying. Your personal Instinct asks before trusting a group, the group agent has no direct access to personal accounts, and pending replies are held when a new member joins [5].

Andrew Chen argues that agents don’t have network effects by default. Better models, UX, memory, and distribution “are not network effects” [6]. He suggests measuring network effects against acquisition, engagement, and monetization KPIs, with density mattering more than scale [6]. He expects a fight over the boundary. Horizontal agents will try to bring artifacts, transactions, and introductions inside their walls, while incumbent networks will try to stay open to every agent. If the agents win, switching agents “means leaving your network behind” [6].

An a16z panel added demand data. About half of Americans report using AI, but only about 4.5% pay for a subscription. Among payers, the top 10% bring in more than half of revenue, and the top 1% spend \$93 a month against a \$25 median [7]. Speakers said platform capability is improving faster than consumers’ willingness to use it, held back by trust in intimate access [7]. Among 1,500+ early adopters of agents, coding is still the top use case [7].

Tools worth knowing

- **Jev** (TypeSafe AI) returns decisions with probabilities: yes/no, one choice from up to 255 options, or a score on 2–10 levels. Input costs \$0.042 per million tokens and output is free [8]. In The Product Compass’s own 50-document test, it made the fewest mistakes and was 115× cheaper than Opus [8].
- **n8n vs. Claude Code.** n8n’s growth has moved from personal automations to business-critical workflows. It presents itself as the orchestration layer for work you can’t trust to run correctly only “95% of the time” [9]. SAP invested at a \$5.2B valuation and embedded n8n in Joule Studio [9].

Craft notes

- **Problems over features.** Marty Cagan’s example is setting a goal of 30-minute employee onboarding and letting the team decide whether that means ten features or none [10]. Prototypes are for learning, products for earning, so “don’t confuse what you vibe code with something you could run your company on” [10].

- **Taste in problems.** Paul Graham says juicy problems have “one main thing you’re trying to solve,” while nasty ones bundle extrinsic constraints [11]. Nasty real-world problems can still pay off: “just don’t choose one by accident” [12].
- **Builder bias.** Shreyas Doshi names three causes: an “I am good at product” identity, moral aversion to marketing and distribution, and virtue-signaling about serving users [13]. On reverse interviews, he says interviewers’ follow-up questions tell you more about a company’s talent than its canned prompts do [14].
- **Support as roadmap input.** One founder runs a weekly 30-minute review of the top five tickets, sorted by frequency. Each gets a doc, UI, or process fix with an owner [15].
- **Hardware PMs** facing supply crunches describe pulling procurement into roadmap discussions earlier [16]. Another commenter describes stretching hardware life cycles from five to seven years [17].

Sources

1. X article by @hnshah
2. X post by @hnshah
3. r/ProductMgmt post by u/TheProductDiary
4. r/ProductMgmt comment by u/HelloTaleproof
5. X post by @noahrshinn
6. X post by @andrewchen
7. The Current State of Consumer AI
8. A New Kind of AI Model: Jev and Quick Wins for PMs
9. n8n CEO: Why we aren’t dead, how to build great AI products, and what I look for in an AI PM
10. HR as a Product | Marty Cagan, NATIVE’26 (legendado em português)
11. X post by @paulg
12. X post by @paulg
13. X post by @shreyas
14. How to Do Reverse Interviews
15. r/startups comment by u/Savantrexs
16. r/ProductManagement comment by u/slickdickrick1
17. r/ProductManagement comment by u/DuskLab