Claude Ship lets AI approve code, and PMs should help write the approval rules

PM Daily Digest

2026-11-28

Claude Ship lets AI approve code, and PMs should help write the approval rules

By PM Daily Digest • November 28, 2026

Anthropic's PR agents raise a product question: which changes can hurt customers? PMs who build faster than their teams, AI-drafted PRDs and agent-written roadmap insights all point to the same thing: speed only helps if someone checks the output.

Claude Ship: who decides what AI can approve?

A Mind the Product video covered Anthropic's Claude Ship, a set of three GitHub agents: - **Code Review** has been generally available since October. Teams choose a quick, standard or deep review, and a review averages $15–25 [1]. - **PR Steward** is in beta. It fixes open PRs after failed tests, conflicts or reviewer notes, and leaves a comment explaining how to undo each change [1]. - **Claude Approvals** is also in beta. A policy file in the repo sets how many humans each part of the code needs to approve, and where Claude can approve on its own [1].

Claude Ship is only for Team and Enterprise customers on GitHub, and it's billed by the token [1].

The speaker urged caution. Anthropic's showcase was a documentation fix, "the safest kind of change that exists." Neither named early customer, Talis Digital or Carvana, describes letting Claude approve PRs by itself [1]. He found no approval error rate in the launch materials. Depending on how a team configures it, a bot-written PR can merge without any human review [1].

He argues that product people should help write the policy file. Engineers tend to judge risk by blast radius, meaning how much code a change touches. But "a 50line change to your checkout flow may be scarier than a 500 line" change

to an internal tool, and a small permissions tweak can expose customer data [1]. His exercise: sort your last 20 merged PRs into changes that could hurt a customer and changes that couldn't, then write down the rule you used. Run the tool in observe mode for a month and compare its decisions with your team's [1]. Observe mode still costs money if the policy relies on Claude's review [1]. Measure reverts, incidents and whether what shipped matched what was asked for, not just the number of merged PRs [1].

The PM who out-ships the dev team

A PM built a feature in 2 days that their offshore team had estimated at "multiple weeks." They now want to build new features themselves and have a developer review them [2]. The most useful replies treated the build as a spec, not a shippable feature. Don't merge it as is, and don't end up owning it. Developers should add tests and check server-side authorization, data isolation between customers, missing indexes and N+1 queries [3]. One commenter suggested a way to diagnose the team: track the days from spec to first working demo. If that number doesn't drop once the team gets a prototype instead of a doc, "it's a team problem, not a spec problem" [4].

Two commenters added warnings. One said PR volume "shot through the roof" without a matching rise in end-user value [5]. Another said PMs who build end up maintaining, and discovery and customer work get dropped [6]. Hiten Shah put the shift this way: teams now have "more capacity to execute than we have ideas worth executing," which requires "a very different kind of discipline" [7].

Where AI helps and where it hurts PM thinking

One PM ran out of their Claude budget, wrote a PRD by hand, and ended up thinking through more scenarios [8]. A commenter explained why. Drafting by hand exposes the spots where you don't know the answer yet, while an AI first draft covers them with "confident sounding filler." The suggestion was to write the problem, the audience and the edge cases yourself, then use AI as "a critic" rather than "the author" [9]. Others pointed to pressure in the opposite direction: at one company, not using AI hurts performance reviews [10].

A related thread raised a governance gap. When roadmap tools let agents create and link insights, how do you know an agent's link shaped a prioritization decision, and how do you undo it [11]? Suggested safeguards: keep agent links in a "suggested" state until a human approves them, show the source on each card [12], and keep an audit trail across tools [13].

Metrics need the same checking. In one case, an AI-written revenue query overstated revenue by about 9% because it included refunds. A reviewer at a client firm finds an error that matters to the business in roughly one of every 8–10 AI-generated reports [14]. The fix: reconcile against a number you already trust, hand-check raw invoices, and write down your MRR and churn definitions

before you ask the model for a number [15].

Quick hits

- **AI role taxonomy.** Aakash Gupta separates three roles. AI PMs ship specs, evals and prototypes. Forward deployed engineers build deployments for a single customer. Product builders write code with AI, then review it before shipping. He says most product builders today hold AI PM titles, and you can move toward the role "from the PM seat you already hold" [16].
- **Shreyas Doshi** introduced "Playing Product vs. Doing Product." He says people with strong product skills can still end up playing product, and it comes down mostly to mindset [17].
- **Paul Graham:** partnerships are "usually a waste of time" for startups. The only kinds worth having are buying someone's product or having them buy yours [18].
- **Hiten Shah:** "There's a much bigger market for making people's lives better than there is for making people better at using AI" [19].
- **Scott Belsky** called Scribe "true enterprise AI PMF." He cites 7M users and more than 90k paying companies, including 48% of the Fortune 500 [20].

Sources

1. Anthropic Launches Claude Ship. Here's How It Works
2. r/ProductManagement post by u/dangflo
3. r/ProductManagement comment by u/darkdali92
4. r/ProductManagement comment by u/darsh-dave
5. r/ProductManagement comment by u/ruraro
6. r/ProductManagement comment by u/Lost_Order6113
7. X post by @hnshah
8. r/ProductManagement post by u/Interesting-Wish4979
9. r/ProductManagement comment by u/TrulyN
10. r/ProductManagement comment by u/[deleted]
11. r/ProductManagement post by u/PomeloGreedy4196
12. r/ProductManagement comment by u/Alternative_Square53
13. r/ProductManagement comment by u/Build_to_scale
14. r/startups post by u/nickvaliotti
15. r/startups comment by u/donvvo
16. substack
17. X post by @shreyas
18. X post by @paulg
19. X post by @hnshah
20. X post by @scottbelsky