Ravi Mehta and Marty Cagan: AI makes building cheap, so PMs have to own the outcome

PM Daily Digest

2026-11-20

Ravi Mehta and Marty Cagan: AI makes building cheap, so PMs have to own the outcome

By PM Daily Digest • November 20, 2026

Ravi Mehta and Marty Cagan argue that AI makes building cheap, so PMs are judged on outcomes and economics. Also covered: AI pricing with no usage data, design sense for PMs, and AI-era tech-debt disputes.

When building is cheap, outcomes are the job

Ravi Mehta defines **Business Outcome Ownership** as linking product improvements to team and company objectives. Every investment has to pass two tests: does it create meaningful customer value, and can that value become value for the company? He also says to stay with the problem after launch, compare the results with what you expected, and decide what to do next [1]. His AI argument: impact used to be capped by how much a team could build. Now outcomes are limited by understanding and judgment, and a team "can… ship more features while creating less value" [1].

Mehta's most useful example is about economics. An AI support feature can grow usage while customers still need several attempts and still contact support, and the company pays for every interaction [1]. Inference, human review, infrastructure and maintenance belong in the investment case. A feature "can win on usage and still destroy margins." That may be worth it only for a specific strategic advantage with a credible path to better economics [1].

Marty Cagan says much the same. Most companies use AI to speed up the project model, and teams get faster without getting better outcomes. He cites this "AI productivity paradox" as noted by McKinsey and Atlassian [2]. Strong teams separate **building to learn** (prototyping to find a solution) from **building to earn** (building something the business can run on and monetize) [2]. As the cost of building to earn falls, "the bottleneck is what do we build?" [2]. On

team shape, he says some teams now have three or four engineers instead of five to ten, while what they own is growing [2]. The PM stays responsible for value and business viability: whether it's affordable, monetizable and compliant [2].

A warning about metrics

A PM at a very large enterprise says a goal to cut median epic cycle time by 15% led teams to clone aging epics and close the old ones. An internal article then celebrated a 35% drop, with no discussion of the real goal of resolving epics within 90 days [3].

Pricing an AI product with no usage history

An r/ProductManagement thread offers practical steps: - **Run an announced trial.** Absorb costs for two to three months and say upfront that usage pricing will follow. Expect some users to "go hog wild" and then scale back, while moderate users get locked in [4]. - **Use COGS as the floor, not the price.** Build forecasts with finance, and pilot with a select group first [5]. For school buyers, set tiers by what a teacher or class needs per term. Model peak demand from the calendar, not from traffic [6]. - **Limit usage by tier.** Put a hard cap on the free tier and soft caps (throttling or overages) on paid tiers. The soft caps show who would pay for more [6].

PMs closer to the build

Aakash Gupta argues PMs need design sense because they are now designing. At AI-native companies they ship to production, and at large enterprises they build AI prototypes [7]. He had Claude Code build a feedback-triage app from one prompt and found three problems. The PRD sat about 470px below the fold. The ranking put a theme with $895K in ARR above one with $1.00M without explaining why. The pasted input took up half the screen [7]. In a separate note he describes Rippling CPO Matt MacInnis's first changes: markdown pushed to a git repo, customer issues found through MCP, and PMs fixing their own copy. Gupta's takeaway is that PMs now need skills such as understanding repo structure and configuring tooling, which used to belong to engineers [8].

A tech-debt standoff with AI-reliant developers

A non-technical product owner says developers will not touch tech debt until the PO writes stories with acceptance criteria, finds and prioritizes the debt, prescribes fixes and sets time limits. The PO's manager wants them at senior-developer level within three months [9]. The developers reportedly say it's a product-owner failure if a story can't be pasted into Claude and produce a first implementation in one pass [10]. Their AI-generated debt list has 7,700 items, such as "Resolve escape: line 33, file XYZ", and they want the PO to rank

them [11]. The PO thinks the developers should own all of this [9], and most commenters agree.

Launch videos

Hiten Shah says Claude Code built a Beam launch video in nine minutes but left out the one scene showing what Beam does [12]. After reviewing eight product videos, his test for each scene is: what does the viewer understand after it [13]? One example is Cursor's video, which reaches "All Checks Passed" and a merge button around 19 seconds in. Developers recognize that payoff, and he would hold on it a beat longer [14].

Sources

1. Business Outcome Ownership.
2. AI-First Isn't Enough: What Winning Product and CX Teams Do Differently w/ Marty Cagan
3. r/ProductManagement comment by u/JohnnyUtah59
4. r/ProductManagement comment by u/clemtiger2011
5. r/ProductManagement comment by u/snoballs_in_daparish
6. r/ProductManagement comment by u/Acceptable-Run-6542
7. Crash Course in Design for PMs
8. substack
9. r/ProductManagement post by u/Naive_Age_4320
10. r/ProductManagement comment by u/Naive_Age_4320
11. r/ProductManagement comment by u/Naive_Age_4320
12. X post by @hnshah
13. X post by @hnshah
14. X post by @hnshah