

AI-First PMs Are Being Asked to Show Working Software—and Defend the Core

PM Daily Digest

2026-09-22

AI-First PMs Are Being Asked to Show Working Software—and Defend the Core

By PM Daily Digest • September 22, 2026

The period's clearest signal is a split: AI is pushing PMs closer to working software and agent-mediated distribution, while behavioral validation, core-product discipline, and auditable product knowledge become more important—not less.

Big Ideas

Agent access is becoming a distribution battleground. Scott Belsky's current framing is that owning the UI remains a moat “for now” even as the UI comes under attack. He describes a new negotiation over “favored agent” status and a “headless services war,” and speculates that restricting consumer-agent access may be part of the transition. [1, 2, 3]

For consumer and service products, map not only the human user flow but also where an agent discovers, authorizes, and completes the job. Treat agent compatibility and access terms as product-strategy assumptions, not integration plumbing to solve after launch.

Tactical Playbook

Turn AI prototypes into behavioral tests, not prettier demos. Sachin Rekhi's method is unusually concrete: make the prototype functional with real data, a database, login, and—where useful—LLM integrations; add PostHog analytics, session replays, and heatmaps; let customers drive it during 1:1 sessions; then send it to dozens of customers with an embedded survey and compare stated feedback with observed behavior. [4]

Use the sequence as a discovery loop: 1. Build the smallest realistic workflow. 2. Instrument it before recruiting users. 3. Observe customers attempting the

job rather than presenting the concept. 4. Define success criteria before the test and plan for false positives and false negatives, as Teresa Torres recommends. [5]

Front-load decisions before AI-assisted development starts. One B2B practitioner recommends aligning with design and lead engineering first, labeling decisions by confidence, and asking the coding agent to expose gaps before the spec is finalized. The point is to prevent the agent from silently making product decisions while the team is already building. [6]

Case Studies & Lessons

Snap and Discord: strengthen the core before chasing side quests. Lenny Rachitsky’s summary of Peter Ellis points to Snap’s response to its 2018 redesign: after DAU flattened, the team focused on performance for existing users—especially on Android—and then saw a multi-year growth renaissance. Discord likewise focused on becoming the best place for intentional multiplayer gaming with friends rather than chasing Midjourney’s moment. [7]

The reusable tool is a **Core Product Value**: one memorable user promise paired with operating metrics. Snap’s promise—“the fastest way to share a moment with the people you care about”—was translated into load time, camera-open-to-share rate, and best-friends engagement; the phrase without metrics is a slogan, while metrics without it are only a dashboard. [7] Use that pairing to test whether a growth initiative improves the product’s central behavior or merely adds activity around it.

Career Corner

“Full Stack PM” is a capability stack, not a replacement for product craft. Aakash Gupta’s model keeps designers and engineers in place but adds AI prototyping and small PRs to the PM toolkit, so the PM can bring working software rather than only a spec. He explicitly says the fundamentals remain: know users deeply, exercise taste about what to add, and direct the build/ship/measure loop toward strategy. [8]

Build one small prototype end to end, but make the career evidence the judgment it enabled: what behavior changed your view, what you stopped, and why the product should ship.

Tools & Resources

Use a Raw → Wiki → Schema research stack. Aakash Gupta’s lightweight knowledge system preserves unedited transcripts, threads, and PDFs as the **Raw** layer; records conclusions once in a linked **Wiki**; and adds a **Schema** so the team can query its evidence instead of rereading it. Skipping the raw layer makes later conclusions unauditably. [9]

For every discovery program, preserve the source material, link decisions to it, and only then structure recurring fields such as segment, problem, evidence, confidence, and outcome.

Sources

1. X post by @mignano
2. X post by @scottbelsky
3. X post by @scottbelsky
4. X post by @sachinrekhi
5. X post by @ttorres
6. r/ProductManagement comment by u/all_happened_before
7. X post by @lennysan
8. substack
9. substack