

AI Coaching, Multiplayer Teams, and Sharper Positioning

PM Daily Digest

2026-04-18

AI Coaching, Multiplayer Teams, and Sharper Positioning

By PM Daily Digest • April 18, 2026

This issue covers three shifts shaping PM craft: AI as a context-rich coach, collaboration patterns that keep teams out of single-player mode, and a refreshed positioning framework. It also includes portfolio triage tactics, segment-focus lessons, career signals, and practical tools worth testing.

Big Ideas

1) AI is becoming a scalable product coach — if you give it the right operating model and context

Foundation models can serve as personal coaches for product creators when they are explicitly configured for the **product model** and loaded with **strategic context** such as product vision, strategy, team topology, and objectives [1]. Marty Cagan's test is practical: the model does not need to match an elite coach; it needs to be good enough to help people improve relative to the average manager most PMs actually have access to [1]. He also expects always-on access to compress PM ramp time from about **three months to closer to one month** [1].

Why it matters: craft coaching has always been scarce, especially as manager span of control rises [1]. This makes product coaching more accessible, but only if the model is used to improve judgment rather than replace it [1].

How to apply: - Tell the model which operating model to follow; avoid a mix of product-model and project-model advice [1]. - Load company-specific context before asking for recommendations [1]. - Ask it to act as a **coach**, not a PRD generator: challenge thinking, focus on outcomes, and avoid empty affirmation

[1]. - Keep final judgment human, especially for non-deterministic outputs and org-specific politics [1].

2) AI can increase output while pulling teams into single-player mode

John Cutler warns that AI can push teams to trust synthesized signals more than colleagues, turning collaborators into internal competitors and amplifying pre-existing problems like high work in progress [2]. His counter is **multiplayer mode**: use AI to spark context sharing and deeper collaboration rather than replacing it [2].

He frames this with the **four E's of cognition** — embodied, embedded, extended, and enactive — arguing that understanding comes from interaction among people, tools, environments, and live dialogue, not isolated document production [2].

Why it matters: a faster artifact is not the same as better shared understanding. Teams can produce more notes, plans, and OKRs while creating less real alignment [2].

How to apply: - Walk through AI-generated prototypes with teammates or users instead of treating the output as self-explanatory [2]. - Use AI to make the work *more visible*, not just cleaner-looking, through information radiators that expose mess and constraints [2]. - Let AI surface patterns from messy discussions, then have the team challenge the synthesis together [2]. - Bias toward multiplayer sense-making until the team agrees on the nature of the problem [2]. - Shift leadership communication from one-way context broadcasts to dialogue, back-briefs, and scenario exploration [2].

3) Positioning is a living answer to ‘why pick us now?’

April Dunford’s updated framing is straightforward: positioning explains how a product is **the best in the world** at delivering value a well-defined customer set cares about, and answers the buyer’s question, why pick us over the alternatives [3]. It is built from five components: **competitive alternatives, distinct capabilities, differentiated value, best-fit accounts, and market category** [3].

Why it matters: positioning should describe the product you have **now** versus the competitors and buyer expectations you face **now**. Strategy drives what you build; positioning explains why the current version wins today [3]. That means positioning may need to change when products evolve, competitors acquire capabilities, market shocks hit, or buyers start asking new questions such as AI strategy [3].

How to apply: - Include the **status quo** in your competitive set, not just obvious software rivals [3]. - Define **distinct** capabilities, not globally unique ones [3]. - Describe **best-fit accounts** at the account level, not as personas [3]. - Test positioning in sales conversations before polishing the website [3].

Tactical Playbook

1) Set up an AI coach or peer that stays grounded

A recurring pattern across PM discussions is that AI advice improves when it has persistent context. One PM building a custom M365 Copilot agent is using a prompt plus persistent files for each product or platform [4], and another practitioner said those files were essential for keeping advice anchored in real constraints rather than generic output [5].

Step-by-step: 1. Start by naming the operating model you want the system to use, such as the **product model** rather than the project model [1]. 2. Load strategic context: vision, strategy, team topology, objectives, or platform files [1, 5]. 3. Instruct the model to act like a coach: challenge assumptions, push on reasoning gaps, and optimize for outcomes rather than making you feel good [1]. 4. Use either a natural-language chat flow or an over-the-shoulder mode that can spot issues you would not think to ask about [1]. 5. Bring org-specific politics, real people, and local power dynamics back to human managers or mentors [1].

Why it matters: this turns AI from a generic answer engine into a reusable thinking partner.

2) Triage an underperforming product without making an open-ended bet

One senior PM described owning **three B2B product lines**: two are healthy, while one has low adoption, weak engagement, dissatisfied customers, and renewal risk tied to real revenue [6, 7]. The core question was whether to pull resources from the healthy products to fix the laggard [6].

Step-by-step: 1. Confirm that the upside is real: customer research should point to meaningful customer and business impact, not just internal frustration [6]. 2. Revisit **discovery basics** before scaling investment: market-problem fit, problem-solution fit, solution-segment fit, product-market fit, plus the problem statement, buyer persona, user persona, and value model [8]. 3. Talk to a small set of users and buyers directly before expanding the bet [8]. 4. Make a narrow **MVP** bet first, using it to derisk the thesis and limit blast radius; the PM's initial framing was to validate within about **three months** [6]. 5. Timebox the effort with explicit success metrics and pull back quickly if the changes do not move the needle [9].

Why it matters: the mistake is not reallocating resources; it is reallocating them without clear kill criteria.

3) Refresh positioning in five steps, then validate it in live selling

Dunford's updated process is shorter, but the prep work matters. Before starting, confirm that you actually have customers, decide whether you are position-

ing the company or a specific product, and align on go-to-market structure such as **land-and-expand** versus suite selling [3].

Step-by-step: 1. Define the real **competitive alternatives**, including status quo behavior like spreadsheets or manual work [3]. 2. List your **distinct capabilities** against those alternatives [3]. 3. Translate those capabilities into **differentiated value** customers care about [3]. 4. Identify **best-fit accounts** rather than jumping straight to personas [3]. 5. Choose the **market category** you want to win [3]. 6. Turn the result into a sales pitch and test it with qualified prospects before updating the website [3].

Why it matters: Dunford’s view is that if the story does not work in live sales conversations, it is not ready for broader messaging [3].

Case Studies & Lessons

1) Todoist shows that frictionless can be audible

Todoist learned that many users capture tasks while driving and cannot look at the screen to confirm anything landed. The team responded by adding **distinct sounds**: one for a new task and another for an edit, so drivers could keep their eyes on the road and still trust the app [10].

“Frictionless doesn’t mean invisible. Sometimes it means audible.”
[10]

Lesson: low-friction UX is about reducing confirmation burden, not removing every signal.

2) When one product serves two segments, pick a market before you split your budget

An edtech founder described a product that solves adjacent pain points for **college students** and **early/mid-career professionals**. Students offered faster scale through institutional partnership but lower willingness to pay without that support; professionals offered higher willingness to pay but slower scale [11].

The community advice leaned hard toward focus: - if the product and core value are the same, the change may be mostly in **marketing language and messaging** [11, 12] - few startups can survive splitting budget across multiple segments early [12] - do not confuse **user** with **customer**; prioritize the segment that can actually buy now [12]

Lesson: the community advice favored early focus over parallel segment bets.

3) Portfolio rescue decisions are easier when you treat them like experiments

The three-product B2B portfolio example surfaced a familiar emotional trap: the data said there was high-impact work to do on the underperformer, but the

PM worried about slowing two healthy products for a fix that might fail [6].

Lesson: when the business case is real but confidence is limited, frame the decision as a **bounded experiment:** - small initial scope [6] - explicit success metrics [9] - a short validation window [6] - willingness to stop if the research thesis does not convert into results [9]

Career Corner

1) Product sense is still the table stakes skill

Cagan argues that the primary use of coaching is developing **product sense** [1]. In his framing, that means learning: - customers and user types [1] - industry and regulatory dynamics [1] - competitive landscape [1] - business data and its levers [1] - **viability**, which can span roughly **5 to 15 dimensions** such as legal, compliance, privacy, monetization, marketing, and sales considerations [1]

How to apply: use AI coaching to accelerate the learning, but keep the target skill human: better judgment about customers, markets, data, and viability [1].

2) AI can accelerate creator growth faster than leader growth

Cagan's split is blunt: creators are mostly dealing with craft, while leaders carry much more politics and situation-specific judgment [1]. He sees strong value for creators, and thinks AI can materially speed development for PMs, designers, and engineers [1].

How to apply: - if you are an individual contributor, use AI heavily for craft development, prototyping, and sense-making [1] - if you are a leader, use it to rehearse scenarios and stakeholder conversations, but keep human coaching close for org-specific dynamics [1]

3) Moving from PO to PM is a scope upgrade, not a gratitude upgrade

One PO described being trapped in small task creation, subtasks, and frequent refinement sessions, with little space left for strategic PM work [13]. The community response was realistic: PM is still not a particularly thanked job, but staying as engineering's task master is a weak long-term bet [14, 15].

How to apply: make the move if it increases strategic ownership, not because you expect more recognition.

4) The field is asking for less prompt theory and more shipping judgment

One experienced PM said there is a **massive gap** between the theoretical prompt engineering taught in AI PM courses and the real work of shipping

products under technical ambiguity [16]. The discovery questions they are collecting are telling: what are the real hurdles in shipping AI features, which tools are actually used daily, and where workflows are still broken [16].

How to apply: build evidence in messy shipping environments, not just familiarity with AI vocabulary.

Tools & Resources

1) A more disciplined Claude Code workflow

Aakash Gupta highlighted a PM workflow where Claude is asked to **interview the user first**, sometimes for up to **30 minutes**, to expose missing angles and unclear goals before writing begins [17]. After that, the user reviews the plan in plan mode, corrects scope, adds phases and checkpoints, and only then approves execution [17]. In the example shared, **six parallel agents** then run research and synthesis, turning what would be a **three-day PM task into about 45 minutes** of execution time [17].

“The single biggest lesson: slow the planning down to speed the output up.” [17]

Worth exploring: the linked Claude Code PM OS [17].

2) Practical AI uses PMs are reporting right now

In a discussion from a hardware PM working in a sensitive industry, one reply listed concrete uses that are already paying off: - deep research for quick topic overviews [18] - generating and reviewing texts, ideas, and presentations [18] - automated or semi-automated bug descriptions and feature-related text handling [18] - small programs or SQL scripts for support and data fixes [18] - rapid prototypes, including ones that use the real codebase to look and feel closer to production [18]

Worth exploring: these examples came from a thread where teams were still dealing with external-tool vetting and limited customer data [19].

3) OFMOS Essential for portfolio-management practice

OFMOS Essential is a tabletop game built around **20 years of research** into commoditization, portfolio evolution, and loss of strategic focus [20]. Players manage **nine products across nine environments** on an **81-position board**, with actions that map to launching, commoditizing, innovating, repositioning, and retiring products [20]. The creator says playtesting quickly surfaced portfolio-level thinking that many PMs rarely practice explicitly [20].

Worth exploring: as a strategy game, a business simulation, or a facilitated learning session with debriefs [20].

Sources

1. Coaching in the Age of AI
2. Single Player to Multiplayer: AI, Context, and Collaboration – John Cutler | ShipSummit | Rise8
3. Obviously Awesome 2.0: What’s Changed in Product Positioning?
4. r/ProductManagement post by u/TechFlameMaster
5. r/ProductManagement comment by u/After_Variation_9567
6. r/ProductManagement post by u/Humble-Pay-8650
7. r/ProductManagement comment by u/Humble-Pay-8650
8. r/ProductManagement comment by u/Bernhard-Welzel
9. r/ProductManagement comment by u/melissaleidygarcia
10. X post by @ttorres
11. r/startups post by u/Much_Basis_6238
12. r/startups comment by u/AnonJian
13. r/prodmgmt post by u/ButterButerfly
14. r/prodmgmt comment by u/Disco_Infiltrator
15. r/prodmgmt comment by u/NeophyteBuilder
16. r/prodmgmt post by u/Sidd_Viscious
17. substack
18. r/ProductManagement comment by u/Mobile_Spot3178
19. r/ProductManagement post by u/squishy_lime23
20. r/prodmgmt post by u/cmitreanu