AI PM Role Maps, Smarter AI Bets, and Parallel Agent Workflows

PM Daily Digest

2026-03-24

AI PM Role Maps, Smarter AI Bets, and Parallel Agent Workflows

By PM Daily Digest • March 24, 2026

This issue maps the AI PM job market, offers a practical framework for deciding when and how to use AI, and highlights two execution shifts: phone-orchestrated agent work and the gap between flashy AI prototypes and product-quality outputs.

Big Ideas

1) AI PM is splitting into clearer lanes

AI PM roles now break across two axes: **traditional PMs adding AI features** versus **AI-native PMs building products where AI is the product**, and **application / platform / infra** layers in the stack [1].

- **What the market looks like:** Traditional PM with AI features is **80% of roles**, while AI-native PM is **20%**. The traditional category has **4x more open roles** [1].
- **Where the technical bar rises:** Application PMs account for **60%** of roles, platform PMs **30%**, and infra PMs **10%**; the deeper the layer, the harder the technical bar [1].

Why it matters: Resume positioning, interview prep, portfolio choices, and target companies change depending on which lane you choose [1].

How to apply: Pick one role type and one stack layer before you start building projects or rewriting your resume. If you are transitioning from a traditional PM background, application roles are the clearest entry point [1].

2) Good AI product strategy starts with saying no

Aakash Gupta's decision rule is simple: use AI for **pattern recognition in complex data**, **prediction from historical data**, and **personalization at scale** [1]. Prefer heuristics or rules when **explainability is non-negotiable**, **clear domain rules exist**, **data is limited**, or **speed matters more than sophistication** [1].

> The best AI PMs know when to say no to AI. That judgment is more valuable than knowing how to build a RAG system. [1]

Why it matters: Teams often over-apply LLMs to problems that would be faster, cheaper, and more reliable with rules or simpler ML approaches [1].

How to apply: Treat whether a problem should use AI at all as the first product decision, not the last. If the answer is yes, match the technique to the job: traditional ML for structured prediction and explainability, deep learning for image/video/audio tasks, and GenAI for conversational, generative, or synthesis-heavy work [1].

3) Non-AI-native startups are now making portfolio-level strategy calls

Andrew Chen notes that many non-AI-native startups funded in the **2020-2025** window are deciding whether to **reinvent the product to be AI-native**, **pivot toward AI**, or **use AI in the back office and ride it out** [2]. His warning: **opportunity cost is the hardest thing to calculate**, and the most dangerous startups may be the ones with just enough revenue to keep going [2].

Why it matters: This is no longer just a feature-roadmap question. It is a company-level product strategy question [2].

How to apply: In annual planning, force an explicit comparison between the cost of reinvention, the cost of a pivot, and the cost of standing still [2].

Tactical Playbook

1) A practical sequence for building AI features

1. **Choose workflow or agent first.** Use a workflow for predetermined, deterministic sequences. Use an agent when the system needs to make decisions, reason, act, and learn across steps [1].
2. **Start with prompts and examples.** System prompts set behavior; few-shot examples show the model what good and bad outputs look like. The source notes that teams can double response quality by adding **3-5** strong examples instead of more instruction text [1].
3. **Engineer context deliberately.** Separate **immediate**, **session**, and **knowledge** context, and load only what the task actually needs [1].

4. **Use RAG before fine-tuning.** For enterprise or domain-grounded answers, chunk documents, convert them into vectors, store them in a vector database, retrieve the nearest matches, and pass those chunks into the LLM [1].
5. **Escalate in the right order.** Optimize **prompts**, then **context engineering**, then **RAG**, and only then consider **fine-tuning**. Gupta's claim is that **80% of use cases** are solved by RAG [1].

Why it matters: It gives PMs a build order that avoids premature complexity and keeps the team focused on the highest-leverage fixes first [1].

How to apply: Turn these five steps into your default review checklist for new AI features.

2) How to set up a parallel AI workbench with Claude Dispatch

1. **Configure desktop first.** Set up Cowork on desktop with the connectors you actually use, such as Gmail, Notion, and Slack, and keep the desktop awake [3].
2. **Start work from mobile.** Open the Claude mobile app, use the Dispatch tab, and ask it to run a Cowork task [3].
3. **Give file access in a usable way.** Grant folder access by describing folders naturally or by using shortcuts; start with the workspace that contains your **CLAUDE.md** and knowledge files [3].
4. **Load your rules before delegating.** Ask Dispatch to read your **CLAUDE.md** before it creates subtasks so the instructions it writes are sharper [3].
5. **Solve file transfer once.** Sync the Cowork workspace folder with Google Drive so files move automatically between desktop and phone [3].
6. **Run tasks in parallel.** From one mobile thread, start multiple independent task sessions, check progress, redirect each one, and bridge context only when needed [3].

Why it matters: The setup matches how PMs actually work across multiple parallel workstreams rather than forcing one-task-at-a-time behavior [3].

How to apply: Use it for work that benefits from breadth and iteration while you are away from your desk: competitor tracking, research synthesis, stakeholder drafts, and visual iteration [3].

Case Studies & Lessons

1) A 48-hour Dispatch test suggests AI can change day design, not just task speed

In one 48-hour experiment, the author directed **60+ task sessions** from a phone while producing competitor summaries, comparison tables, sponsor pages, gap analyses, and multiple infographic iterations [3]. The reported split was

roughly **25 minutes** of human direction versus **3+ hours** of parallel Claude execution [3]. The author's summary of the work split: **90% human thinking**, **100% human takes and opinions**, and **90% Claude research and formatting** [3].

> Use AI to amplify your thinking, not to replace it. [3]

Why it matters: The lesson is not just faster output. It is that async direction from a phone can reshape how a PM structures the day [3].

How to apply: Keep judgment, prioritization, and opinion with the PM; let AI take the first pass on research, drafting, and formatting [3].

2) Fast AI prototypes still miss the work that makes a product usable

Sachin Rekhi argues that **AI prototyping is easy to start and hard to master** [4]. His critique of many one-prompt prototypes is specific: they may look impressive at first, but often **do not match the design of the existing product**, **lack meaningful differentiation**, and **fail to master the core workflows** [4]. His response is an **AI Prototyping Mastery Ladder** with **15 essential skills** [4].

Why it matters: Speed to a functional demo can hide whether the prototype is actually good product work [4].

How to apply: Review prototypes against three gates before you get excited: design fit, differentiated value, and quality on the core workflow [4].

Career Corner

1) The best AI PM entry path is narrower than it looks

For PMs trying to break into AI, the highest-volume lane is still **traditional PM with AI features**, which represents **80% of roles** and roughly **4x** the openings of AI-native roles [1]. Within the stack, **application PM** roles are **60%** of the market and are described as the easiest entry point for someone moving from a traditional PM background [1].

Why it matters: You do not need to target the hardest, deepest roles first to get into AI PM [1].

How to apply: If you are transitioning, aim first at traditional-plus-application roles, then deepen toward platform or infra once you have shipped AI work [1].

2) Hiring managers want shipped products and a portfolio that proves range

Gupta's advice is to **build products, not projects**: launch, get real users, and learn from what breaks [1]. He recommends three portfolio artifacts with real users:

- a product solving a real problem you have [1]
- an agent that demonstrates goal-oriented reasoning [1]
- a RAG system grounded in a domain you know well [1]

Why it matters: This portfolio shows both general product execution and AI-specific judgment [1].

How to apply: Replace tutorial clones with artifacts that show users, failure modes, fixes, and product decisions [1].

3) Evals and company environment are becoming career signals

Gupta frames AI evals in a simple structure: **inputs**, a **task that generates outputs**, and a **scoring function from 0 to 1** [5]. He also says the **AWS AI Practitioner** certificate can complement hands-on work, but certification alone is not enough [1]. And he highlights that different company cultures train different PM muscles: **Amazon** emphasizes writing and customer-backwards docs, **Meta** emphasizes experimentation, and **Netflix** emphasizes autonomy [1].

Why it matters: PM candidates increasingly need to show production thinking and to choose environments that develop the skill they want most [1, 5].

How to apply: Add eval design to your portfolio, pair any certification with shipped work, and be intentional about the PM culture you want to learn in [1, 5].

Tools & Resources

- AI PM at Netflix, Amazon and Meta - Here's How to Become an AI PM (Fundamentals + Job Search) — a useful role taxonomy, AI decision framework, and job-search roadmap for PMs moving into AI [1]
- The Claude Dispatch Guide: 48 Hours Running AI Agents From My Phone — practical setup, workflow examples, and lessons from running PM tasks in parallel across phone and desktop [3]
- Cowork on your desktop — the prerequisite setup guide before using Dispatch [3]
- The AI Prototyping Mastery Ladder — a deeper resource on the 15 skills Rekhi says matter for moving from flashy prototypes to product-quality outputs [4]
- RAG vs fine tuning guide — helpful if your team is comparing prompt optimization, context engineering, RAG, and fine-tuning [1]
- **Claude surface selection:** use **Dispatch** for mobile orchestration of desktop tasks, **Channels** for bidirectional and scheduled work inside active sessions, and **Web Sessions** for remote coding or prototyping [3]
- **Knowledge layer pattern:** store **CLAUDE.md** plus templates, workflows, and knowledge files in a GitHub repo so the system compounds

across surfaces; the claim is that PMs who build this layer can ship at **5x** the pace of ad-hoc users [3]

Sources

1. AI PM at Netflix, Amazon and Meta - Here's How to Become an AI PM (Fundamentals + Job Search)
2. X post by @andrewchen
3. The Claude Dispatch Guide: 48 Hours Running AI Agents From My Phone
4. X post by @sachinrekhi
5. substack