Beyond Accuracy, Better AI Workflows, and a Sharper PM Job Search

PM Daily Digest

2026-04-01

Beyond Accuracy, Better AI Workflows, and a Sharper PM Job Search

By PM Daily Digest • April 1, 2026

This issue centers on practical PM frameworks: a four-part scorecard for GenAI products, a simple screen for deciding what to automate, and concrete playbooks for Claude workflows, support-to-docs loops, and narrative clarity. It also includes Amazon case metrics, growth ideas from QR codes, and sharper job-search tactics for PMs.

Big Ideas

1) Evaluate GenAI products beyond accuracy

> Accuracy is a trap. [1]

Accuracy describes model performance, but not whether users trust the product, find it useful, return to it, or whether it creates business value [1]. The framework described in the Product School session evaluates GenAI products across **trust, usefulness, adoption, and business impact** [1]. That matters because a product can be reliable but useless, useful but risky, or well-used but economically unsustainable [1].

How to apply - Make each AI feature prove itself on all four dimensions, not just model quality [1] - Give each dimension concrete metrics, owners, and review

cadences before launch [1]

Evaluating GenAI Products Beyond Accuracy | Amazon AI Product & Technology Leader (2:15)

2) Use a two-question screen before automating PM work

Sachin Rekhi's heuristic is simple: ask whether a workflow is **worth building** and **possible to build** with AI [2]. It is worth building when AI has a clear advantage, such as synthesizing customer interviews faster and more comprehensively, or when the task is frequent and time-consuming, such as weekly status updates [2]. It is possible to build when AI can access the right context, the work can be broken into discrete steps, and human judgment is limited enough that the workflow will not stall [2].

How to apply - Start with recurring PM tasks where AI already outperforms manual effort on speed or coverage [2] - Reject automations that depend on hidden context or undefined judgment calls [2]

3) QR codes are becoming a measurable offline growth channel

QR codes can connect packaging, receipts, events, and out-of-home placements to product experiences with very little friction [3]. The more interesting shift is measurement: dynamic tools such as **ME-QR** let teams update links without reprinting, track sources, segment traffic, and run experiments, effectively bringing performance-style analytics into offline surfaces [3]. The recurring failure modes are basic but important: no clear reason to scan, weak mobile UX,

and no tracking [3].

How to apply - Use QR only when it clearly makes a user job easier; onboarding, retention, support, referrals, and promos are the cited use cases [3] - Treat offline scans like any other channel: instrument source, segment traffic, and test destinations [3]

Tactical Playbook

1) Roll out a GenAI evaluation system in five steps

1. **Week 1:** define the top three metrics per dimension, set baselines, and choose realistic and stretch targets [1]
2. **Weeks 2-4:** instrument the product, set up dashboards, establish human evaluation, and build feedback collection into the experience [1]
3. **Weeks 5-8:** run a pilot with 50-200 users, gather quantitative and qualitative data, and iterate on the gaps [1]
4. **Post-launch:** monitor trust and safety daily, engagement weekly, business impact monthly, and review the product comprehensively each quarter [1]
5. **Keep iterating:** use A/B tests, user feedback, and updated evaluation criteria as the product changes [1]

Why it matters: the speaker's lesson is that pilot data should drive launch decisions, and multi-dimensional evaluation surfaces issues that accuracy alone misses [1].

2) Add learning, memory, and evaluation to Claude with three CLAUDE.md blocks

The Product Compass article proposes three blocks that make Claude more useful for product work: a **Knowledge Architecture**, a **Decision Journal**, and a **Quality Gate** [4].

How to apply 1. Before each task, review domain rules and hypotheses; after each task, store learnings in `/knowledge/{domain}/knowledge.md`, `/hypotheses.md`, and `/rules.md`, and maintain a `/knowledge/INDEX.md` [4] 2. Promote a hypothesis to a rule only after 3+ confirmations, and demote it if new data contradicts it [4] 3. Before major choices, search prior decisions; if none exists, log the decision, context, alternatives, reasoning, trade-offs, and any superseded choice in `/decisions/YYYY-MM-DD-{topic}.md` [4] 4. Add explicit evaluation criteria outside the generation step, because agents tend to praise their own work even when quality is mediocre [4]

Why it matters: after one month, the author reports Claude was automatically applying 24 project-specific rules, and the decisions with three written alternatives were right 80% of the time [4].

3) Turn repeated support questions into documentation work every week

A simple Friday workflow from Lenny's Newsletter: review resolved support tickets, and if a question appeared **3+ times** that week, flag it as a docs or FAQ candidate, create a Linear issue assigned to `@agent`, and include the standard answer as the starting point [5].

Why it matters: it converts recurring support questions into docs or FAQ candidates and ready-to-assign issues [5].

How to apply - Set a weekly review cadence, not an ad hoc one [5] - Use the recurrence threshold to reduce noise and focus only on patterns [5] - Include the existing answer so documentation starts from something already working in support [5]

4) Pressure-test your product story in 20 minutes

> Open a blank document and write down your company's story in 200 words. What job are customers hiring you to do? Why does your approach work when others fail? [6]

Why it matters: Hiten Shah frames this as a basic clarity test, and says most founders cannot do it in the allotted time [6].

How to apply - Limit yourself to **200 words** and **20 minutes** [6] - Answer only two questions: the customer job and why your approach works better than alternatives [6] - Use the exercise as a quick internal clarity check [6]

Case Studies & Lessons

1) Amazon's AI assistant companion: trust features and utility metrics moved together

In the AI assistant example, every response included source attribution, some responses included confidence scores, and human evaluation kept the hallucination rate below **2%** [1]. On usefulness and adoption, the related prompt library drove **40% faster prompt creation**, about **85% thumbs-up**, **3x higher engagement** for manager-specific prompts, **2x retention** for users with community prompt access, and **85%+ returning users** within a few months [1].

> It's a game changer for my workflow and results. [1]

Key takeaway: trust mechanisms such as source attribution are more valuable when they are paired with clear evidence that the product saves time and keeps users coming back [1].

2) Amazon's B2B purchase guardrails: business impact was measurable quickly

The purchase guardrails example generated **several millions in annualized revenue**, served **thousands of business customers**, reduced manual budget tracking by **80%+**, and reached positive ROI within **3 months** [1].

Key takeaway: when an AI product is tied directly to completing a workflow faster and with less manual tracking, PMs can measure business impact in revenue, productivity, and ROI rather than relying on model-centric metrics alone [1].

3) A structured Claude workspace improved through use

The Product Compass author says that after a month, Claude had generated and was automatically applying **24 project-specific rules** extracted from patterns across dozens of sessions [4]. The same write-up says the decisions the author felt most confident about had the worst hit rate, while decisions where three alternatives were written down were right **80%** of the time [4].

Key takeaway: persistent knowledge capture and explicit alternatives can beat confidence-based decision-making [4].

Career Corner

1) PM job search is shifting from volume to precision

Aakash Gupta argues that mass-applying with AI does not work. His recommendation is to apply to fewer, **surgically targeted** roles, stack referrals before submitting, and run the search in about **20-30 minutes a day** instead of three hours [7].

How to apply - Build the referral path before the application: Gupta says cold application callback rates are around **2-4%**, while warm intros are **5x** higher, and every candidate he coached into a top-company offer had a referral on file before the resume went in [8] - Send **25 personalized connection requests per week**, rotate across target companies, follow up on days 3, 7, and 14, and ask for the referral only after context is established [8]

2) Tailored resumes only help if they stay truthful

Gupta's warning on AI resumes is blunt: many tools either invent experience or produce generic keyword swaps, and invented experience can backfire when interviewers check it [8]. His recommended standard is a JD-specific resume built only from real experience [7].

How to apply - Restructure the resume around the specific job description, but only with evidence you can defend in interview [7, 8] - Treat fabrication as a risk, not a shortcut [8]

3) Specific work products and interview prep still create separation

Gupta highlights a **90-minute** work product: a one-pager analyzing the company's product plus a working prototype of the recommendation [7, 8]. He also emphasizes company-specific prep, including interview formats, reported questions, and screening signals across **250 companies**, plus mock interviews that identify weak areas over time [7, 8]. On the back end, he recommends negotiation research and counter-offer drafts because the compensation impact can be meaningful [7].

How to apply - Use a work product when a standard application is not creating enough signal, but make it specific enough that it could only have been written for that company [8] - Build interview prep around the target company's actual format and questions, not a generic PM script [7, 8]

Tools & Resources

1) The CLAUDE.md blocks from Product Compass

What it is: a reusable set of three blocks for learning across sessions, logging decisions, and evaluating output quality [4].

Use it for: ongoing product domains where patterns emerge slowly, teams re-debate the same choices, or AI output needs a separate quality bar [4].

2) Prompt patterns from Lenny's Newsletter

What they are: ready-made automation prompts for PLG lead qualification, recurring support-to-docs conversion, and launch management [5].

Use them for: workflows with clear cadence and routing rules. The examples also show when specialization helps: **Sage** handles course operations and reminders, while **Kelly** checks Linear daily, starts a branch, and opens a PR for assigned dev tasks [5].

3) Dynamic QR tools such as ME-QR

What it is: a way to change destinations without reprinting codes, track sources, segment traffic, and run experiments from offline touchpoints [3].

Use it for: packaging, receipts, events, support, referrals, and promo mechanics where you want a measurable bridge from offline to product [3].

4) An AI prototyping checklist from r/ProductManagement

What it covers: the integration layer between LLM APIs, vector databases, and preprocessing; state and context handoffs in RAG systems; token-cost monitoring; and the practical shift toward CLIs for Claude workflows [9, 10].

Use it for: early planning before a PM-led prototype or side project so the first blockers are visible before implementation starts [9].

Sources

1. Evaluating GenAI Products Beyond Accuracy | Amazon AI Product & Technology Leader
2. X post by @sachinrekhi
3. r/prodmgmt post by u/Low-Sir-8366
4. Three CLAUDE.md Blocks That Make Claude Get Smarter Every Session
5. OpenClaw: The complete guide to building, training, and living with your personal AI agent
6. X post by @hnshah
7. The Claude Code Job Search Operating System
8. substack
9. r/ProductManagement comment by u/Flimsy_Actuator_6947
10. r/ProductManagement comment by u/TheKiddIncident