

# Decision Stacks, Outcome OKRs, and the New Quality Bar for AI Products

PM Daily Digest

2026-04-23

## Decision Stacks, Outcome OKRs, and the New Quality Bar for AI Products

*By PM Daily Digest • April 23, 2026*

This brief focuses on the product systems that are becoming more important at the same time: strategy-to-execution alignment, outcome-based measurement, and a higher quality bar for AI products. It also includes practical discovery, monetization, stakeholder-management, and career takeaways from Robinhood, Stripe, and the broader PM community.

### Big Ideas

#### 1) Build a connected decision system, not isolated strategy docs

Martin Eriksson's **Decision Stack** frames alignment as five linked questions: where are we going, how will we get there, what matters now and how do we measure progress, what actions will we take, and how do we choose between them. His suggested layers are **vision, strategy, objectives and key results, opportunities, and principles**. He positions it as a mental model rather than a rigid framework, so teams can keep existing tools as long as the layers connect [1].

The core benefit is connection: leadership sets direction, teams bring bottom-up insight, and the stack narrows options so teams stop re-litigating the same tradeoffs. That matters because context-free empowerment creates drift, and one cited stat is stark:

“95% of employees do not know their organization strategy.” [1]

- **Why it matters:** Better alignment is not about more documents. It is about making vision, strategy, objectives, opportunities, and principles reinforce each other so teams can make faster decisions with less debate [1].

- **How to apply:** Start by auditing what already exists, define terms together, review strategy at least quarterly, integrate the stack into existing ceremonies, and start small instead of trying to create the whole system at once [1].

## 2) Restore OKRs to outcomes

Several sources converge on the same warning: OKRs fail when they become renamed roadmaps, copy-paste cascades, or individual performance tools [2]. A strong objective is qualitative and aspirational; a strong key result is quantitative and measures **human behavior change**, not a feature, launch date, or project milestone [2]. Cascading should be a critical-thinking exercise about what a team can influence, not a mechanical exercise where a parent KR becomes a child objective [2].

“fall in love with the problem, not the solution.” [2]

- **Why it matters:** Output-shaped OKRs let teams stay busy without proving value. Outcome-shaped OKRs force a clearer link between customer behavior and business impact [2].
- **How to apply:** If current planning is too rigid, split OKRs into **discovery, build, and outcome** types, work backward from fixed dates earlier, and cut the metric set down to the handful of difference-makers that truly map to top objectives [2].

## 3) AI leverage is moving from assistance to transformation, but the quality bar is rising with it

Sachin Rekhi’s **AI leverage continuum** is a useful lens: **Assist** uses AI as an input to a larger task, **Automate** hands AI an end-to-end workflow, and **Transform** redesigns the task itself around AI’s new capabilities [3]. His product-development example sits at the transform end: prototype many ideas first, release them internally, and prioritize the ones that resonate because AI has made code cheaper to produce [3].

A separate SaaS discussion adds the market reality: established companies are at risk if they ship AI features that are only “**60% solutions**” because users can often build something better themselves with AI coding tools [4]. The same source argues that complex agentic products still need forward-deployed onboarding help, that **stealth churn** can show up in usage drops before revenue drops, and that agent-friendly APIs are becoming a retention issue because AI tools make APIs accessible to many more people [4].

- **Why it matters:** AI strategy is no longer just “add an assistant.” It touches product development, onboarding, retention, and platform design [3, 4].
- **How to apply:** For each initiative, ask whether you are assisting, automating, or transforming. Then audit whether the product is actually

good enough to beat a DIY alternative, whether onboarding needs human help, whether usage is slipping silently, and whether your API passes a simple agent integration test [3, 4].

#### 4) Quality still needs rituals, not taste alone

Stripe’s product-quality practice is to “**walk the store**”: everyone is expected to test end-to-end journeys and look for dead ends or mismatches across products. The company also tracks a subset of **essential journeys** on a red/yellow/green scoreboard and reviews these experiences publicly on Fridays so different disciplines can spot different issues [5].

“fight the gravitational pull to mediocrity and do not leave well enough alone” [5]

- **Why it matters:** Fast-moving products drift at the seams. Cross-product journeys can degrade even when individual teams think their own area is fine [5].
- **How to apply:** Define the journeys that matter most, review them on a visible scoreboard, evaluate prototypes the way an uninformed user would experience them, and ship a **minimum viable quality product** so you can learn without losing trust [5].

### Tactical Playbook

#### 1) Run a manual validation sprint before writing code

A founder exploring a plant-diagnostics SaaS noticed that subreddit posts routinely lacked basic diagnostic context like watering frequency, drainage, soil type, and symptom timing. Before building, they spent **three weeks** answering questions manually with the same structured intake and no product mentions [6].

- **Why it matters:** Manual service can reveal whether the problem is real before you commit to a product. In this case, the founder got a **70% reply rate** and an average of **12+ follow-up questions** per thread, then concluded that follow-up questions were a better signal than upvotes [6, 7].
- **How to apply:**
  1. Pick a problem space where requests are frequent but context is missing [6].
  2. Create a fixed intake with the same questions every time [6].
  3. Answer manually for a set period without pitching a product [6].
  4. Track reply rate and follow-up depth, not just attention signals [6, 7].
  5. Only then decide whether the workflow deserves software [6].

## 2) Reboot OKRs with a lighter operating model

- **Why it matters:** Teams often force every piece of work into one OKR shape, then wonder why planning becomes dishonest or brittle [2].
- **How to apply:**
  1. Label current work as **discovery**, **build**, or **outcome** instead of pretending everything is already an outcome [2].
  2. Work backward from hard dates earlier, when decisions can still be relaxed instead of rushed [2].
  3. Set OKRs only for the part of the world your team can influence [2].
  4. Make key results about behavior change, not feature shipment [2].
  5. Prune aggressively; one organization cut **341** key results down to about **40** measures that mattered [2].

## 3) Speed up regulated launches by moving compliance into the core team

Robinhood's advice is simple: bring legal and compliance in early, make them feel like owners of the product, and solve rules as product constraints instead of treating them as blockers [8]. The company also says its 2022 move from a functional structure to a GM model helped speed decisions because product, engineering, compliance, and operations were rolled into one org instead of negotiating across silos [8].

- **Why it matters:** In regulated environments, late-stage compliance review slows shipping and weakens product quality [8].
- **How to apply:** Bring partners in at the concept stage, align on the vision together, and give one team shared responsibility for the outcome rather than separate functional veto points [8].

## 4) Monetize advanced insight features without surprise paywalls

In one startup discussion, a founder wanted to keep data entry and pattern visibility free while putting a more advanced analysis page behind a subscription [9]. The strongest community advice was consistent: keep some insight free, charge for deeper analysis, and frame the paid layer as an upgrade from day one rather than taking away something users thought was permanent [10, 11].

- **Why it matters:** Long-gestation products need users to experience value before paying, but sudden removal creates backlash [9, 10].
- **How to apply:** Keep basic insights free, offer advanced analysis as a time-boxed trial with previews of what users will unlock, and research pricing tiers around user sophistication instead of defaulting to one plan for everyone [10, 11, 12].

## Case Studies & Lessons

### 1) Robinhood: measure commitment, not just activity

Robinhood says its strategy rests on three pillars: be **#1 in active traders**, **#1 in wallet share for the next generation**, and **#1 global financial ecosystem** [8]. Two leading indicators matter especially: recurring **net deposits**, which signal trust, and **Robinhood Gold** subscriptions, a **\$5/month** paid plan that often leads users into more products across cash, retirement, credit, and trading [8].

The company pairs that strategy with a **barbell** UX approach for both new and advanced users, a focus on two or three “magical moments” per product, and AI embedded in support, stock-move summaries, scanners, and an in-app assistant [8]. It also says AI has compressed some early ideation and alignment work from **four to five weeks** to as little as **two to three days** [8].

- **Lesson:** Leading indicators of commitment are often more useful than raw usage, and AI is most credible when it is embedded in existing workflows instead of bolted on [8].
- **How to apply:** Pick one or two commitment metrics, decide which moments in the experience deserve handcrafted excellence, and make AI solve a real job inside the product rather than adding generic novelty [8].

### 2) Stripe: turn quality into a shared operating habit

Stripe’s “walking the store” practice exists because users experience the company as one connected system across products like subscriptions, payments, and tax, even when teams are organized separately [5]. The company reinforces that view with essential-journey scoreboards and Friday walkthroughs where founders demo live and multiple functions review the same experience together [5].

- **Lesson:** Quality improves faster when the whole company can see the same broken journey, not just when a local team reviews its own feature [5].
- **How to apply:** Create a visible journey list, assign ownership, and review real user flows cross-functionally instead of relying on slideware or isolated QA passes [5].

### 3) Williams Sonoma modeling exercise: prioritize AI bets by value, certainty, and speed

In a case-interview exercise built around Williams Sonoma, the retailer was described as an **\$8B** premium home retailer with about **40%** of annual volume concentrated in an **8-week** holiday window, and leadership preferred options with faster payback and lower fixed-cost risk [13]. The exercise considered customer service, selling/conversion, personalization, and forecasting, then prior-

itized customer service and selling as the highest-ROI, fastest-impact options [13].

The modeled comparison was concrete: a customer-service agent handling **5M** chats annually at **60%** resolution implied about **\$48M** in savings, while a styling assistant improving conversion implied about **\$28M** in incremental profit [13]. The recommendation in the exercise was to **buy** a customer-service solution for speed to market, then mitigate differentiation, lock-in, and scalability risks through customization, swappable model APIs, and stress testing [13].

- **Lesson:** AI prioritization is stronger when it combines ROI, certainty, and time-to-value instead of defaulting to the flashiest use case [13].
- **How to apply:** For each AI idea, model cost reduction, revenue lift, pay-back period, and opportunity cost before debating build vs. buy vs. partner [13].

#### 4) BBC Maestro: context can surface better ideas from anywhere

BBC Maestro used the Decision Stack to turn a young company’s finance-heavy planning into a clearer, more shareable articulation of vision and strategy across the business [1]. In a separate example from the same discussion, a junior developer who had been given full strategic context suggested an **80/20** solution that senior leaders had missed [1].

- **Lesson:** Empowerment works better when teams get context, not just autonomy [1].
- **How to apply:** Share the reasoning behind bets early enough that people close to the work can challenge or simplify them [1].

## Career Corner

### 1) AI PM roles are expensive, scarce, and interviewing differently

One data point from the AI PM market: OpenAI reportedly pays AI PMs **\$860K** total compensation versus **\$325K** at Amazon for the same title, while foundation-model companies outpay application-layer companies by **40–80%** [14]. The same source argues scarcity is a major driver because **60%** of AI PMs do not come from CS backgrounds and the pool of people who understand both product and model development is still small [14]. Hiring loops are also tightening, with **4–10** interview rounds and an **AI product design** round emerging as the top failure point [14].

The good news is that internal mobility appears real: the same source says internal moves into AI PM happen in a median of **21 months**, and **12,000+** people transitioned into AI PM roles between January 2024 and October 2025 [14].

- **Why it matters:** The market is rewarding AI PM capability, but not just generic PM strength. Product design for AI products is becoming a

separate screen [14].

- **How to apply:** Build fluency in both product thinking and model-adjacent tradeoffs, prepare explicitly for AI product design interviews, and consider internal transfers if the external market feels closed [14].

## 2) Side PM consulting works best when the expertise is narrow

In a PM community thread on freelance product roles, the consistent view was that “part-time” product work often expands beyond **8 hours a week** because product context takes years to build [15, 16]. The more credible path is advisory or consulting work grounded in deep domain expertise, not generic PM-for-hire positioning [16, 17]. Some commenters were also skeptical that certain postings were really PM jobs at all, suggesting they might be ways to train AI systems on PM work [18, 19].

- **Why it matters:** Side income is possible, but product work is hard to compress unless the client is paying for narrow expertise rather than broad product ownership [16].
- **How to apply:** Favor advisory work in industries where you already have credibility, scrutinize part-time job scopes carefully, and be cautious with roles that look more like knowledge extraction than product leadership [16, 17, 18, 19].

## Tools & Resources

### 1) The Decision Stack

A reusable alignment template built around five questions: destination, path, current priorities, actions, and decision principles [1].

- **Why it matters:** It lets teams map what already exists instead of forcing a full strategy reset [1].
- **How to apply:** Audit current documents, connect the layers, and start with the most broken part of the stack first [1].

### 2) Discovery / Build / Outcome OKRs

A planning template that separates exploratory work, delivery milestones, and true outcomes instead of forcing everything into one bucket [2].

- **Why it matters:** It creates a more honest operating rhythm for teams that are discovering, shipping, and launching at the same time [2].
- **How to apply:** Label current work first, then tighten definitions over time rather than throwing out the roadmap wholesale [2].

### 3) The AI Leverage Continuum

A simple framework for classifying AI work as **Assist**, **Automate**, or **Transform** [3].

- **Why it matters:** It helps PMs distinguish incremental AI features from operating-model changes [3].
- **How to apply:** Review your roadmap item by item and ask whether each initiative is merely helping a task, automating it, or redefining it [3].

#### 4) Essential Journeys Scoreboard

Stripe’s red/yellow/green tracking system for the subset of user journeys that matter most [5].

- **Why it matters:** It keeps cross-product quality visible instead of burying it in team-local dashboards [5].
- **How to apply:** Pick the journeys users depend on most, review them in public, and invite multiple functions into the walkthrough [5].

#### 5) The “Agentic API” test

A practical heuristic from the SaaS discussion: ask a vibe-coding tool to build a simple integration or dashboard against your API, then see how quickly it works [4].

- **Why it matters:** If agents and non-developers struggle to use your API, churn pressure rises as easier integrations win [4].
- **How to apply:** Run the test on your own product and on vendors you depend on; if the integration is painful, treat it as a product problem, not just a developer complaint [4].

---

### Sources

1. Why 95% of employees can’t name their organisation’s strategy - Martin Eriksson (The Decision Stack)
2. Episode 267: How OKRs Become Outputs Instead of Outcomes
3. X post by @sachinrekhi
4. Managing 20+ AI Agents: Lazy Agents, Our \$500K AI Bill, Stealth Churn & the Death of 60% Solutions
5. How Stripe Built Their New Website
6. r/startups post by u/ByTheSea1969
7. r/startups comment by u/ammie12
8. Robinhood VP of Product on Winning Wallet Share for Gen Z & Millennials
9. r/startups post by u/rosettacoin
10. r/startups comment by u/cutie-patootie-427
11. r/startups comment by u/Final\_Boat\_9360
12. r/startups comment by u/AnonJian
13. Consulting case interview: AI strategy case (w/ BCG and A&M Consultants)

14. substack
15. r/ProductManagement post by u/anotherhappylurker
16. r/ProductManagement comment by u/aslittatti
17. r/ProductManagement comment by u/anotherhappylurker
18. r/ProductManagement comment by u/NaCheezIt
19. r/ProductManagement comment by u/audaciousmonk