

Intent-led roadmaps, mid-delivery “Why” checks, and feedback-driven regression tests for AI agents

PM Daily Digest

2026-02-23

Intent-led roadmaps, mid-delivery “Why” checks, and feedback-driven regression tests for AI agents

By PM Daily Digest • February 23, 2026

This edition focuses on three execution-level levers: using roadmaps to communicate intent (not feature lists), keeping “Why” questions alive during delivery, and building regression protection for AI agents by turning customer feedback into test cases. It also includes a real-world career transition scenario (dev → PO) with practical positioning angles.

Big Ideas

1) Roadmaps are a leadership tool—communicate intent, not a feature list

A product roadmap can function as a bridge between long-term vision and day-to-day execution, working best when it communicates *intent* rather than just timelines ¹. The underlying shift is toward **outcomes over outputs**—using the roadmap to keep teams focused on solving meaningful problems instead of shipping more features ².

Why it matters: Roadmaps often fail when they become task lists or “feature factories,” which can detach delivery from strategy ³⁴.

How to apply: Define the outcome you’re trying to drive first (vision → outcomes), then ensure roadmap items stay tied to that intent ⁵⁶.

¹[r/prodmgmt](#) post by [u/Straight_Ad8809](#)

²[r/prodmgmt](#) post by [u/Straight_Ad8809](#)

³[r/prodmgmt](#) post by [u/Straight_Ad8809](#)

⁴[r/prodmgmt](#) post by [u/Straight_Ad8809](#)

⁵[r/prodmgmt](#) post by [u/Straight_Ad8809](#)

⁶[r/prodmgmt](#) post by [u/Straight_Ad8809](#)

2) “Why” doesn’t have a closing time—keep discovery questions alive during delivery

Many teams treat “Why” as a phase (discovery → alignment → execution), but the premise here is that this is an expensive assumption: constraints surface, learnings emerge, markets shift, and the “right problem” can change midstream ⁷. Teams that ship what matters keep asking whether they’re still solving the right problem and moving toward the intended outcome—even when work is already moving fast ⁸.

Why it matters: “Fast-work assumptions” (clear owner, faster decisions, alignment, locked requirements, on-time delivery) can crowd out the courage to ask tough questions at the moment they’re most needed ^{9,10}.

How to apply: Make “Why checks” part of execution—small interruptions to prevent long detours ¹¹.

3) For AI agents, treat customer feedback as regression tests

When an AI agent is handling revenue-critical conversations, regressions from prompt changes are a core risk ¹². One approach: convert every piece of customer feedback into a test case, and have the agent rerun the conversation until it passes—building an accumulating test suite that brings “code testing” rigor to LLM behavior ¹³.

Why it matters: This creates a mechanism to protect what already works while iterating—paired with a measurable operational impact (customer review dropping from 100% to 5% of conversations) ¹⁴.

How to apply: If you’re shipping agent changes frequently, make “feedback → test case” an always-on loop, not a one-time QA effort ^{15,16}.

Tactical Playbook

1) Pick a roadmap format based on the alignment problem you’re solving

Different roadmap formats address different needs:

⁷When Did You Last Ask Why?

⁸When Did You Last Ask Why?

⁹When Did You Last Ask Why?

¹⁰When Did You Last Ask Why?

¹¹When Did You Last Ask Why?

¹² post by @ttores

¹³ post by @ttores

¹⁴ post by @ttores

¹⁵ post by @ttores

¹⁶ post by @ttores

- **Now–Next–Later** to stay agile without locking into unrealistic deadlines ¹⁷
- **Timeline-based** when you need alignment across departments or external stakeholders ¹⁸
- **Theme-based** to connect work to strategic outcomes and OKRs (and avoid “feature factory” planning) ¹⁹

Step-by-step: 1. **Name the primary constraint:** do you need agility, cross-functional alignment, or strategy-to-work traceability? ²⁰ 2. Choose the format that matches that constraint (Now–Next–Later / Timeline / Themes) ²¹. 3. Keep the roadmap strategic—so plans don’t get mistaken for commitments ²².

2) Run a “mid-delivery Why check” (10 minutes) to avoid months of wrong execution

This perspective argues that asking “Why” midstream isn’t going backwards—it prevents arriving somewhere nobody wanted to go ²³.

Step-by-step (questions to use): 1. Re-anchor on the problem: “What problem are we actually trying to solve?” ²⁴ 2. Make the customer change explicit: “What changes for our customers once we ship this?” ²⁵ 3. Validate outcome direction: “Are we still moving toward the outcome we said we cared about?” ²⁶ 4. Pressure-test the bet with recency: “Would we make the same bets today that we made three weeks ago?” ²⁷

Why it matters: The claim is leverage: “Ten minutes of Why” can save “three months of How” ²⁸.

3) Prevent roadmap over-promising by planning for learning and change

A recurring failure mode: over-promising dates. Estimates change, and a roadmap should evolve as you learn from users, experiments, and feedback ²⁹.

Step-by-step: 1. Start with vision and intended outcomes—otherwise even polished roadmaps degrade into task lists ³⁰. 2. Involve stakeholders early to

¹⁷[r/prodmgmt post by u/Straight_Ad8809](#)

¹⁸[r/prodmgmt post by u/Straight_Ad8809](#)

¹⁹[r/prodmgmt post by u/Straight_Ad8809](#)

²⁰[r/prodmgmt post by u/Straight_Ad8809](#)

²¹[r/prodmgmt post by u/Straight_Ad8809](#)

²²[r/prodmgmt post by u/Straight_Ad8809](#)

²³[When Did You Last Ask Why?](#)

²⁴[When Did You Last Ask Why?](#)

²⁵[When Did You Last Ask Why?](#)

²⁶[When Did You Last Ask Why?](#)

²⁷[When Did You Last Ask Why?](#)

²⁸[When Did You Last Ask Why?](#)

²⁹[r/prodmgmt post by u/Straight_Ad8809](#)

³⁰[r/prodmgmt post by u/Straight_Ad8809](#)

reduce misalignment later ³¹. 3. Review the roadmap regularly so it reflects reality, not wishful thinking ³². 4. Keep it strategic rather than overly detailed to avoid confusion between “plans” and “commitments” ³³.

Case Studies & Lessons

1) ShowMe: converting customer feedback into a durable QA system for AI agents

In the context of revenue-critical sales conversations, ShowMe’s approach is to automatically turn each piece of customer feedback into a test case ³⁴. The agent reruns the conversation until it passes; over time this builds a battery of tests so new prompt changes don’t break what already works ³⁵.

Outcome metric: customer review reportedly drops from 100% of conversations to 5% ³⁶.

Takeaway: Treat feedback as an asset that compounds into coverage—not as isolated anecdotes ³⁷³⁸.

2) Roadmap failures often come from date over-commitment, not lack of detail

A stated lesson: roadmap failures commonly come from over-promising dates; a strong roadmap evolves as learning accumulates, staying strategic instead of overly detailed ³⁹.

Takeaway: Use the roadmap to signal intent and direction, then update it as user feedback and experiments change what you know ⁴⁰⁴¹.

3) Career transition case: developer moving toward Product Owner work for higher leverage

A mid-level Android developer with ~10 years of experience and prior business ownership (running a gaming server company) describes deriving more value from PO/Scrum Master work—prioritizing, clarifying requirements, aligning

³¹_r/prodmgmt post by u/Straight_Ad8809

³²_r/prodmgmt post by u/Straight_Ad8809

³³_r/prodmgmt post by u/Straight_Ad8809

³⁴ post by @ttores

³⁵ post by @ttores

³⁶ post by @ttores

³⁷ post by @ttores

³⁸ post by @ttores

³⁹_r/prodmgmt post by u/Straight_Ad8809

⁴⁰_r/prodmgmt post by u/Straight_Ad8809

⁴¹_r/prodmgmt post by u/Straight_Ad8809

business and devs, making tradeoffs, shipping, and strategizing—than from coding itself ⁴². They’re willing to take an estimated ~40% pay cut to start as junior/mid PO, and would even intern unpaid ⁴³.

Takeaway: The transition tension is often about *positioning* (no official PO title despite doing the work) rather than motivation ⁴⁴.

Career Corner

1) Positioning for a PO move when you don’t have the title (but you’ve done the work)

A practical framing from the transition case: the challenge is avoiding the perception of “dev who’s bored of coding,” despite having PO-like experience and stakeholder understanding ⁴⁵.

How to apply (using only what’s in the case): - Describe your experience through **PO deliverables** (prioritization, requirements clarity, alignment, tradeoffs, shipping, strategy) rather than through role labels ⁴⁶. - If you’ve run a business, explicitly include that you’ve operated with real tradeoffs and outcomes (the case includes two years running a gaming server company) ⁴⁷. - Be direct about the reset you’re willing to take (junior/mid PO, pay cut, internship) so hiring managers understand your expectations and commitment ⁴⁸.

2) Use “Why questions” as a leadership signal—not a phase-gate ritual

The “Why has no closing time” argument can double as a career skill: being the person who keeps the team oriented around the right problem and outcome during delivery, not only in kickoff/retro moments ⁴⁹⁵⁰.

How to apply: Bring a lightweight set of Why prompts into active work (problem, customer change, outcome direction, and whether you’d still make the same bet) ⁵¹⁵²⁵³⁵⁴.

⁴²[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴³[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴⁴[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴⁵[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴⁶[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴⁷[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴⁸[r/product_design](#) post by [u/Ok_Piano_420](#)

⁴⁹When Did You Last Ask Why?

⁵⁰When Did You Last Ask Why?

⁵¹When Did You Last Ask Why?

⁵²When Did You Last Ask Why?

⁵³When Did You Last Ask Why?

⁵⁴When Did You Last Ask Why?

Tools & Resources

- **Building Effective Product Roadmaps (Product 360 blog)** — a full write-up referenced alongside the roadmap takeaways ⁵⁵.
 - **Productify: “When Did You Last Ask Why?”** — the source essay arguing Why should persist through delivery, with practical question prompts ⁵⁶⁵⁷.
 - **Product 360 (tool mention)** — cited as a tool that can help turn strategy into visual, collaborative roadmaps (with the caveat that the “real work” is the mindset shift to outcomes over outputs) ⁵⁸.
-

Sources

1. r/prodmgmt post by u/Straight_Ad8809
2. When Did You Last Ask Why?
3. post by @ttorres
4. r/product_design post by u/Ok_Piano_420

⁵⁵r/prodmgmt post by u/Straight_Ad8809

⁵⁶When Did You Last Ask Why?

⁵⁷When Did You Last Ask Why?

⁵⁸r/prodmgmt post by u/Straight_Ad8809