

From Agent Evals to Glass-Box Enterprise AI

PM Daily Digest

2026-08-01

From Agent Evals to Glass-Box Enterprise AI

By PM Daily Digest • August 1, 2026

Supabase’s agent benchmark and Decagon’s productized deployment model point to a new PM bar: measure real workflows, turn implementation into reusable product, and preserve human taste.

Big Ideas

Agent compatibility is moving from claim to test. Supabase introduced Evals, running Claude Code, Codex, and Open Code against real tasks and scoring what they do. [1] Paul Graham says he expects all services used by agents eventually to adopt such tests—and that services agents cannot use could go out of business. [2] For PMs, define “agent-ready” through representative workflows and measurable success criteria; turn failed runs into product priorities.

Fast AI cycles favor stable direction over detailed schedules. Decagon says a precise 12-month roadmap is difficult when build cycles are so fast; it keeps themes and a clear long-term vision, lets customer signals determine what to build, and retains human judgment over what to include or exclude. [3] Apply this as “stable vision, short commitment horizons”: make the current bet explicit, keep later bets provisional, and review them against fresh customer evidence.

Tactical Playbook

Make feedback earn a roadmap slot. A community practitioner’s process is: stop accepting feature requests and capture the problem or symptom; quarterly observe 10–15 users in their normal environment; collect about 15 pains; have 100–150 users rank them; prioritize with a weighted average; and reserve roughly one-third of sprint capacity for customer-satisfaction fixes, leaving two-thirds for strategy and technical debt. [4] Use those figures as a starting hypothesis, not a law. At minimum, track each theme’s source, segment, observed behavior, frequency, severity, and blocked business outcome; promote it only when evidence

is strong, and link the roadmap item to the evidence and to what would change your mind. [5]

Separate “Now” from discovery. One startup stopped weekly roadmap churn by keeping Now stable except for critical or regulatory items while allowing Next and Later to change; the commenter says the approach held through acquisition and scale-up. [6] The organizational prerequisite is role clarity: define what PM owns before hiring, and keep early teams on high-trust, light rituals rather than importing frameworks and OKR cascades too soon. [7]

Case Studies & Lessons

Decagon’s “glass box” turns deployment into product. Its forward-deployed teams are expected to contribute to core product, so a capability built for one enterprise becomes available to the next 10 customers rather than remaining one-off work. [3] In a customer comparison, a Sierra deployment produced about three new journeys over a year because the customer depended on forward-deployed engineers; with Decagon’s productized model, the same customer created about seven in a month, with nontechnical teams able to act directly. [3] Lesson: treat recurring implementation labor as product debt. Instrument what specialists repeatedly do, generalize it, and give customers enough control to iterate without waiting on the vendor.

Career Corner

PM interviews are becoming build tests. The discussion describes companies replacing presentation rounds with live prototypes to assess “full stack builders.” [8] Evaluators look for taste and decision ownership—not a prototype where AI made all the choices—and flag generic copy, weak underlying data, and designs that feel like AI output. [8] Prepare by giving the tool context in three buckets—functionality, design, and data—then attach a wireframe and realistic dataset; add a live API and be ready to explain front/back-end boundaries, security, model choice, and caching. [8]

Tools & Resources

Choose the prototype stack by the job. The current tool map separates design-system/front-end tools (Reforge Build, Magic Patterns, Alloy), full-stack zero-to-one tools (Lovable, Bolt, Replit), and full AI development tools (Claude Code, Codex). [8] For interview work, the recommended differentiators are visual context, thoughtful copy and data, and a working API—not faster generic generation. [8]

Sources

1. X post by @supabase
2. X post by @paulg
3. Decagon's Playbook for Building Enterprise AI Applications
4. r/ProductManagement comment by u/GeorgeHarter
5. r/ProductManagement comment by u/coolreddy
6. r/ProductManagement comment by u/LakeHold
7. r/ProductManagement comment by u/rash3rr
8. Prototyping in Interviews | Theory & Practice