

Designing for Agents, New Users, and Better PM Judgment

PM Daily Digest

2026-06-11

Designing for Agents, New Users, and Better PM Judgment

By PM Daily Digest • June 11, 2026

This brief highlights new PM lessons on agentic product design, interface-led growth, and the workflows that matter as models improve. It also covers concrete tactics for discovery, instruction-file audits, coaching, and self-serve PM automation.

Big Ideas

- **AI products need an autonomy spectrum, not a single mode.** Linear sees three user groups: non-users, people who want AI with approval, and people willing to fully delegate. Its triage feature uses historical routing data to classify and route incoming bugs or requests while still letting customers choose how much control to keep [1]. YC adds a complementary constraint: when AI changes the economics of a workflow, redesign the process end to end - but keep the product surface area small and bounded [2]. **Why it matters:** PMs now have to define the human/agent handoff explicitly. **Apply it:** map one workflow into manual, approve, and delegate modes before adding more automation.
- **The biggest growth bets may come from new interfaces, not smarter models.** OpenAI's early fit was strongest in knowledge-worker-heavy markets like Germany and the US, but much weaker in Brazil and India [3]. Search broadened everyday usefulness, and image generation opened ChatGPT to people less likely to use a text-first interface. India became OpenAI's #2 market, and Image Gen 2 launched at 1,512 ELO, about 240 points above the next competitor [3]. **Why it matters:** deeper intelligence and broader adoption are different roadmap jobs. **Apply it:** force every major bet into one of two buckets - deepen current users, or

unlock people who cannot use the product today.

Tactical Playbook

1. **Interview for the signal the model does not have.** YC argues customers rarely hand you the winning prompt; they describe a local optimum shaped by their own constraints [2]. A startup example with 3,080 users and only one paid conversion shows the right next step: interview the payer on why they bought and a cross-section of free users on why they did not, then test packaging or paywall changes from there [4, 5, 6]. **Why it matters:** execution is cheaper, but hidden demand is not. **Apply it:** compare payer vs. non-payer decision paths, capture willingness-to-pay language verbatim, and decide whether you have a painkiller or a vitamin before changing the roadmap.
2. **Re-audit your AI instruction files when the model gets better.** *The Product Compass* argues that old CLAUDE.md files, duplicated rules, drifted facts, and guardrails written for weaker models can actively hold back stronger ones [7]. **Why it matters:** better models can inherit worse habits from legacy instructions. **Apply it:** ask the model to review its own instructions before you edit them, then cut contradictions and stale rules. Default effort to *high*, reserve *max* for rare cases, and use */goal* patterns for long unattended PM work [7].

“Don’t fix anything yet. Report first. I decide what gets cut.” [7]

Case Studies & Lessons

- **Linear is moving from issue tracker to “product development system for teams and agents.”** The shift includes optional but default-ready agentic workflows: triage incoming feedback, create issues or PRDs from transcripts and notes, and connect third-party or internal agents through APIs across tools like Slack, Gong, and Intercom [1]. Messaging has also moved upmarket from feature language toward value language and customer proof points [1]. **Takeaway:** centralize context, then let automation meet users where they already work.
- **Brex’s AI rethink started upstream, not at the task level.** Instead of only building an agent for KYC, the team redesigned onboarding end to end. That moved risk qualification earlier in the funnel, making it possible to KYC leads rather than only customers and changing who they target [2]. **Takeaway:** when AI makes a downstream task cheap, revisit upstream qualification, targeting, and process boundaries.

Career Corner

- **PMs are becoming faster adopters of agentic workflows.** Linear says non-engineering roles - especially PMs - have made some of the biggest

recent gains, often using agents for self-serve work like meeting-to-issues or PRD drafting instead of waiting on engineering or data partners [1]. **Apply it:** start with one repeatable workflow where the output is easy to review, not one where the model becomes the decision-maker.

“Coaching is not about telling people what to do or giving them answers. It’s about holding a space and reflecting...” [8]

- **Use coaching to improve judgment, not outsource it.** Mind the Product describes most PM coaching relationships as a coach/mentor hybrid, with the client still responsible for the decision [8]. Good sessions start with a current blocker or frustration, and peer triads can work well inside organizations [8]. LLMs can help with structured reflection, but not replace human accountability [8]. **Apply it:** spend 5-10 minutes before a coaching session naming the behavior or decision you want to change.

Tools & Resources

- **Keep the instruction-file audit prompt handy.** It is a practical template for cleaning up PM agent rules before your next model upgrade [7].
- **Try a lightweight LLM accountability loop.** A morning agenda prompt plus end-of-day recalibration in Slack helped one coach stay focused and reduce shiny-object drift [8].

Sources

1. Linear COO on Rebuilding the Product Dev Lifecycle for Teams and Agents | Cristina Cordova | E299
2. The CEO Must Be the Chief AI Officer
3. substack
4. r/startups comment by u/edkang99
5. r/startups comment by u/oalbrecht
6. r/startups comment by u/avtges
7. The Ultimate Guide to Claude Fable 5
8. How to get the most out of coaching in product