Context Graphs, Habit Loops, and Git Workflows for AI-Native PMs

PM Daily Digest

2026-07-02

Context Graphs, Habit Loops, and Git Workflows for AI-Native PMs

By PM Daily Digest • July 2, 2026

This brief covers two important PM ideas: designing products around internal triggers and building enterprise AI systems that learn from shared decisions. It also includes a concrete Git workflow for PM-owned AI artifacts plus a few practical resources for building AI fluency.

Big Ideas

- **AI needs shared memory, not just chat.** Asana argues enterprise AI spend showed **0% productivity gains** in cited Goldman Sachs and McKinsey research because people were using chat tools individually, copying data in and out, while decisions never made it back into a shared context graph [1]. Their counter-model is a work graph that captures goals, portfolios, projects, and tasks—"who does what by when and how"—so approvals, corrections, and process data can compound into better agent behavior over time [1]. **Why it matters:** PM teams will get more leverage from systems that retain decisions than from isolated prompts. **How to apply:** route PRD reviews, launch plans, and customer-feedback decisions into one shared system, and make approvals/rejections visible so future agent outputs can improve.

  > "…none of those decisions are making it back into a context graph that will create that compounding benefit." [1]

- **Habit formation starts with internal triggers, not notifications.** Nir Eyal's Hook Model remains a useful frame for frequent-use products: **trigger → action → variable reward → investment** [2]. The key shift is from external triggers like pings to internal triggers like boredom, loneliness, or uncertainty [2]. The final step matters because each use

should improve the product, with AI making more "market of one" customization feasible [2]. **Why it matters:** it gives PMs a concrete test for whether a product can become a habit. **How to apply:** use it for high-frequency products, not low-frequency ones like insurance [2].

Tactical Playbook

1. **Version PM-owned AI artifacts in Git.** Aakash Gupta's note is simple: PMs should version **skills and evals** the same way other teams version important files [3]. The working rhythm is straightforward: pull the latest changes, create a branch, edit, commit with a descriptive message, push, open a PR, and merge after review [3]. **Why it matters:** prompt-like assets change constantly; Git gives you review, history, and rollback. **How to apply:** start with one evaluator or reusable skill, then move it into a repo with PR-based review.
2. **Separate AI for personal speed from AI for team execution.** Asana describes two modes: headless/MCP access through tools like Claude, ChatGPT, and Gemini for fast individual data access [1], and an in-graph multiplayer mode where role-based AI teammates watch work, respond across people, and take action inside the shared system [1]. **Why it matters:** not every AI workflow belongs in chat, and not every workflow needs shared automation. **How to apply:** use headless tools for quick retrieval and updates; reserve shared-agent workflows for recurring, cross-functional processes.

Case Studies & Lessons

- **Fitbod shows the Hook Model in practice.** Eyal describes uncertainty at the gym as the internal trigger; opening the app is the action; the unknown workout, reps, sets, and weight create the variable reward; and logged workouts improve future recommendations [2]. He says the team confirmed they built the app from the Hook Model [2]. **Takeaway:** habit loops are strongest when every use makes the next session better.
- **Asana moved PLG and forward-deployed AI talent into product.** The company moved its PLG team—including pricing, packaging, experimentation engineering, and PM resources—under product, with a GM who owns revenue and reports to the CPO [1]. It also put an incubation team of forward-deployed AI engineers under the Asana AI GM so early customer learnings feed directly back to PMs and engineers [1]. In one workflow, voice-of-customer requests are reviewed by an AI agent, routed to a PM for judgment, and then sent to an AI coder that generates a PR for engineering review [1]. **Takeaway:** if AI changes acquisition or delivery, shorten the loop between customer signal, PM judgment, and implementation.

Career Corner

- **The PM skill gap on agents is widening, but the advice is practical.** Product Compass says PMs are splitting into two speeds: those who work with agents every day and those who have not started yet [4]. Their recommendation is not more theory: start with VS Code, where **90%** of the learning transfers to any agent, and learn by doing, breaking things, and fixing them [4]. **How to apply:** pick one recurring PM task this week—analysis, drafting, or prototyping—and run it end to end with an agent inside VS Code.

Tools & Resources

- **The AI-Native PM Roadmap**: weekly **90-minute** live sessions at **6:00 PM CET**, with demos, materials, homework on your real product, and recordings afterward [4]. The first three sessions are free and move from Cowork to Codex to Claude Code in VS Code [4]. Later sessions, recordings, exercises, and a certificate option are for premium members [4].
- **GitHub for PMs**: the full playbook co-written with Shubham Saboo, including the rollback workflow for PM-managed AI assets [3].

Sources

1. Asana CPO on Why Every Employee Is Now an AI Eval | Arnab Bose | E302
2. How to hack back your attention span & take control of your mind | Nir Eyal on Tetr Podcast Ep2
3. substack
4. The AI-Native PM Roadmap: A Live Session Every Week, First 3 Free