

Prototype-First PM, Dynamic Workflows, and Better v1 Decisions

PM Daily Digest

2026-06-08

Prototype-First PM, Dynamic Workflows, and Better v1 Decisions

By PM Daily Digest • June 8, 2026

This brief covers the move from PRD-heavy planning to prototype-led work, a concrete AI workflow for turning interviews into ranked opportunities and testable concepts, and Tony Fadell's lessons on judgment, iteration, and storytelling.

Big Ideas

- **Prototype-first PM is moving from exception to default.** Aakash Gupta argues the PRD is no longer the main output; it is an input to a prototype. Docs force design, engineering, and legal to simulate different versions of the product, which can create false alignment. A prototype gives everyone the same object to react to. **Why it matters:** faster, sharper feedback earlier. **Apply it:** build the smallest working or clickable artifact you can, then attach a short FAQ covering the hypothesis, V1 requirements, edge cases, and success metrics. [1]
- **For v1 products, judgment still matters more than clean-looking data.** Tony Fadell argues that net-new categories have too few analogs for purely data-driven decisions, so a very small group needs to make informed opinion-based calls. He points to the iPhone keyboard debate as a case where the data showed pros and cons on both sides, and Steve Jobs still chose a direction. Fadell's broader point: leaders should micromanage the few details that matter most, while thinking across the whole system—product, distribution, installation, sales, and marketing—not just the UI. **Why it matters:** early data can create precision without differentiation. **Apply it:** decide which decisions need centralized judgment, and evaluate the full experience, not just the feature. [2]

Tactical Playbook

1. **Use dynamic workflows when the next step depends on the last step.** The Product Compass describes these as short JavaScript programs Claude writes on the fly to coordinate agents. Use plain subagents for one round of parallel work; use a dynamic workflow when outputs need to route, score, filter, retry, or verify later stages. [3]
2. **For interview synthesis, turn discovery into a six-stage loop:**
 - extract structured opportunities, personas, and verbatims from each interview
 - canonicalize overlapping needs into a shared set
 - score opportunities by frequency \times importance \times (5 - satisfaction)
 - generate solution ideas and rank them by ROI
 - build the top three as clickable HTML prototypes
 - inspect failures or low-confidence outputs, then rerun only the affected stages [3]
3. **Let code handle coordination, and models handle judgment.** In the worked example, the workflow used 113 agents, 1.95M tokens, and 12.5 minutes to produce 3/3 built-and-verified prototypes; the routing, scoring, gating, and looping logic used zero model tokens. [3]
4. **Automate the plumbing only after alignment.** In one Reddit discussion, PMs stressed that prioritization is still the core job: finance gaps are only one input among many, and they would not automate the decision straight into feature creation. The same thread noted that once teams agree on the metric, tools like Claude Code or even Power Automate can pull finance data and post updates to Jira—but finance access controls may block broad integrations. [4, 5, 6, 7]

Case Studies & Lessons

- **A faster way to surface disagreement:** Abhi Muchhal, a PM at OpenAI, was asked to write a PRD for a platform investment, stopped after 20 minutes, built the thing instead, and found the discussion around the working prototype much better than the doc would have been. The lesson is not to eliminate writing; it is to demote the long spec and let the prototype do the heavy lifting. [1]
- **Most products need multiple generations.** Fadell's rule is simple: make the product, fix the product, then fix the business. He says the iPod did not truly break out until the third generation, when Windows support and the iTunes music store helped it move beyond early Mac loyalists. **Takeaway:** do not judge product-market fit or business viability from version one alone. [2]

Asked how you know whether you are building something people actually want

[8], Hiten Shah’s answer was blunt:

“They will literally tell you once you have.” [9]

Paul Graham adds the founder version of that idea: people who start by making something they themselves want are often better at convincing users than investors. [10]

Career Corner

- **Own the story, not just the backlog.** Fadell argues that builders need to meet customers in their own context, which makes marketing and storytelling part of product work, not a downstream handoff. He says launch communication should distill to a few tentpole features, and he watched Steve Jobs refine the iPhone story repeatedly until the why was clear. **Career takeaway:** if you want broader product influence, practice positioning, messaging, and launch narrative with the same rigor you use on specs. [2]

Tools & Resources

- **Claude Code dynamic workflows / ultracode:** worth exploring if you regularly triage inbound work, synthesize customer calls, or audit large sets of stories. The reusable patterns called out were classify-and-act, fan-out-and-synthesize, adversarial verification, generate-and-filter, tournament, and loop-until-done. PM use cases included synthesizing 100 interviews and checking 80 user stories against INVEST criteria. [3]

Sources

1. substack
2. Father of the iPod and iPhone on building taste, judgment, and creativity in the AI era
3. Claude Dynamic Workflows for PMs: The Ultimate Guide
4. r/ProductManagement comment by u/Forgethestamp
5. r/ProductManagement comment by u/panconquesofrito
6. r/ProductManagement comment by u/waybetter
7. r/ProductManagement comment by u/ITORD
8. X post by @Tristanrhee3
9. X post by @hnshah
10. X post by @paulg