

Prototype-Literate PMs, Healthier Metrics, and Practical AI Workflows

PM Daily Digest

2026-03-28

Prototype-Literate PMs, Healthier Metrics, and Practical AI Workflows

By PM Daily Digest • March 28, 2026

Why prototype-building is becoming part of the PM baseline, where AI actually creates leverage, and how to tighten execution with better metrics, post-mortems, and onboarding design. Also includes hiring signals and tool recommendations PMs can test now.

Big Ideas

1) Prototype literacy is becoming a core PM skill

“Instead of doing some case study and presentation, you need to be ready to build a full blown app as part of the interview.” [1]

Elena Verna argues that functional prototyping is becoming standardized across roles, not just PM [1]. She also says turning a PRD into an interactive artifact improves the PRD itself, helps sell the idea faster, and gives engineers and designers a clearer shared vision [1].



Why PMs who can't build will get left behind / Elena Verna / Product in Practice (6:11)

Why it matters: The expected PM artifact is expanding from documents to clickable experiences. In Verna's workflow, the written spec is intentionally kept to a one-pager, with more detail discovered through prototyping [1].

How to apply: Write the shortest useful spec, prototype it immediately, and use the gaps you find to tighten the hypothesis before engineering starts. Keep engineering in the ideation loop rather than treating the handoff as closed [1].

2) AI gives PMs leverage unevenly: strongest in critique, synthesis, analysis, and execution hygiene

The Exponent framework splits PM work into vision, strategy, design, and execution [2]. In that model, AI is already useful for customer insight synthesis, AI-moderated interviews, natural-language data analysis, prototyping, meeting agendas and summaries, and critiquing an existing strategy [2]. Verna frames the same pattern another way: AI can do the first 30-50% of baseline work across PRDs, prototypes, and marketing plans, so PMs react and refine instead of starting from a blank page [1].

Why it matters: The fastest gains are in compressing time-to-insight and time-to-artifact, not outsourcing the hardest judgment calls [2].

How to apply: Use AI to research, critique, summarize, query data, and draft

prototypes. Do not outsource direct customer contact, product vision, or the final strategic bet [1, 2].

3) North star metrics need a pressure test before they turn into local optimizers

Run the Business offers four meta-questions to pressure-test north star metrics: Would you be proud of the behavior in 18 months? Can the team explain in one sentence how the metric makes a customer's life better? If every team's north star sat on one page, would any of them compete? What happens if you 10x the metric—would that be incredible or terrible? [3]

Why it matters: The framework is explicitly designed to catch anti-patterns before a metric starts driving the wrong behavior or creating cross-team conflict [3].

How to apply: Run these four questions in your next review. If the customer-value sentence is weak or a 10x outcome sounds bad, treat that as a metric-design problem, not just a reporting issue [3].

Tactical Playbook

1) Use AI as a strategy critic, not a strategy author

1. Create a project with explicit devil's-advocate instructions and tell the model not to be nice [2].
2. Load opinionated strategy best practices into project knowledge, such as course material or a strategy book summary [2].
3. Paste in your strategy and ask for critique [2].
4. Look for concrete issues like an audience that is too broad or a claimed moat that is not actually defensible [2].
5. Rewrite the strategy yourself; AI can critique the bet, but it is not the owner of the bet [2].

Why it matters: The demo shows critique quality improves when the model is grounded in a specific standard of good work, not a generic prompt [2].

How to apply: Save this as a reusable review step before leadership reviews. If the output feels generic, add exemplars rather than more vague instructions [2].

2) Build a weekly insight loop from surveys, NPS, and customer data

1. Upload raw survey or NPS data to Claude and ask for the basics first: score, promoter/passive/detractor mix, trends, and segment cuts [2].
2. Add segmentation fields you already have, such as email type, plan type, or usage intensity [2].
3. Check significance where the analysis provides it [2].

4. Turn the output into an executive readout if needed, including an AI-generated deck in Gamma [2].
5. Once you trust the workflow, move from occasional reporting to a recurring cadence [2].

Why it matters: In the example, work that previously took about a week became fast enough to support weekly reporting instead of quarterly review [2].

How to apply: Start with one recurring customer metric and three segments. Expand only after you can verify the numbers and logic [2].

3) Improve AI data analysis with example pairs

1. Let PMs ask data questions in plain English and inspect the generated SQL when needed [2].
2. Expect early errors when schemas are messy or outdated [2].
3. Give the model past natural-language question and SQL pairs as project knowledge [2].
4. Reuse the same exemplar pattern for other PM work, including interview guides and strategy critique [2].

Why it matters: Rekhi's example is not just faster query writing; it expands the number of questions a PM can realistically ask from a few to all of them [2].

How to apply: Build a small internal library of approved examples, then graduate from ad hoc queries to scheduled dashboards [2].

4) Make post-mortems blameless but operational

“Work the problem” [4]

1. Name the single core reason the launch failed [4].
2. Add three supporting reasons or evidence showing how that problem appeared in data or feedback [4].
3. Use screenshots or other specifics to show what happened [4].
4. Model accountability by taking blame yourself so others can do the same [4].
5. End with process changes for next time, without overreacting to one incident [4].

Why it matters: This was shared as a practical structure for leadership-facing post-mortems, where the goal is reuse and learning rather than blame [5, 4].

How to apply: Use the structure as the spine, then connect any survey data or stakeholder feedback back to the core reason and its supporting evidence [5, 4].

5) When something ships broken, own the conversation without creating a product-vs-engineering culture

PMs are often the first stop for bugs, delays, and quality issues from stakeholders and customers [6]. The discussion emphasizes owning the conversation rather than deflecting, avoiding a product-vs-engineering culture, giving the team credit when things go well, and taking blame when they do not [6, 7, 8].

Why it matters: Public ownership builds trust and keeps momentum, while quality issues can also signal deeper team problems [8].

How to apply: Acknowledge the issue, state the recovery plan, protect the team externally, and then use the failure to inspect process or coordination problems internally [6, 8].

Case Studies & Lessons

1) Onboarding by doing beat onboarding by explaining

One PM spent about 40% of initial development time on a polished onboarding with animations, progress indicators, and tooltips, yet day-1 retention was 21% because users skipped through the flow, reached the main app confused, and left [9]. Rebuilding the first-use experience so users performed the core action lifted day-1 retention to 44% [9].

Lesson: Users can complete onboarding without learning anything. What mattered in the follow-up comment was core-task success on day one, not tutorial completion [10].

How to apply: Watch the first session, find the first meaningful action, and redesign onboarding around doing that action instead of explaining it first [9, 10].

2) Faster analysis changed the team's learning cadence

In the NPS demo, a CSV upload produced score summaries, monthly trends, segment comparisons, and statistical significance checks [2]. Because the analysis became much faster, the team could move from quarterly NPS review to continuous weekly reporting and get more fresh insights [2].

Lesson: AI changes operating cadence when the bottleneck is analysis time [2].

How to apply: Look for recurring insight work that is currently too slow or too rare, and automate the end-to-end workflow rather than just one step [2].

3) Spec, prototype, and GTM draft can now happen in parallel

Verna's nonprofit feature demo starts with a ChatGPT tech spec intended to get 30-50% of the way to a usable draft [1]. She then turns it into a one-pager [1], generates a prototype in Lovable and spends the next couple of hours editing

structure, content, and visuals [1], while ChatGPT works in parallel on ICP, distribution partners like TechSoup, and examples to tear down such as Google and Slack nonprofit programs [1]. She says this gets her to spec, prototype, and GTM thinking in roughly three hours, with engineering pulled into ideation rather than a final handoff [1].

Lesson: The power is parallel drafting plus human reaction, not expecting a one-shot answer [1].

How to apply: Run product, UX, and GTM thinking in parallel, then use human taste to edit hard and involve engineering before priorities are locked [1].

Career Corner

1) The job market is improving, but not evenly

PM openings are at the highest levels seen in more than three years [11]. At the same time, Bay Area importance is rising, remote opportunities are declining, and recruiter demand is surging as a leading indicator of sustained hiring demand [11, 12].

Why it matters: Better hiring volume does not mean an easier search if your location or remote requirements are narrow.

How to apply: Broaden Bay Area-targeted searches if possible, reset expectations on remote-only roles, and watch recruiter activity as a sign that demand is holding [11, 12].

2) The skills that still compound are not the ones AI can average away

Verna's non-automated list is direct customer interaction [1], setting the vision and destination [1], understanding marketing and distribution as software commoditizes [1], and building functional prototypes [1].

Why it matters: Her warning is explicit: if everyone uses AI to choose direction, products converge [1].

How to apply: Protect time for customer calls and social listening, build a stronger point of view on where the product should go, and learn enough GTM and prototyping to turn that point of view into something concrete [1].

3) AI-native experimentation is becoming a hiring signal

Verna recommends bringing AI-native employees into teams, often including new grads, because they already treat AI as a normal part of work [1]. She also argues that teams need bottom-up tool adoption and repeated experimentation because model performance changes quickly [1].

Why it matters: The advantage is not one favorite tool; it is an operating habit of trying, judging, and retrying workflows as the tools improve [1].

How to apply: Show concrete AI workflows in your portfolio or resume, push for lightweight experimentation on your team, and revisit tasks that did not work a month ago [1].

4) Senior leadership means more delegation and more accountability

“When the team succeeds, it’s their fault. When we fail, it’s my fault.” [7]

Tony Fadell’s management advice is to let go of doing the work yourself, trust the team, and give people room to be creative [13]. The Reddit thread adds the complementary leadership behavior: be the circuit breaker first when something goes wrong [8].

Why it matters: Advancement is not just better judgment. It is creating space for the team to do great work while absorbing external pressure yourself [13, 8].

How to apply: Delegate real ownership, resist the urge to re-do the work, and take the first uncomfortable conversation when outcomes disappoint [13, 8].

Tools & Resources

1) Perplexity Computer for deliverables, not just answers

Aakash Gupta argues that Perplexity’s Computer produces finished outputs: research reports with citations, deployed dashboards, cleaned datasets with charts, and launch kits with positioning docs and email drafts [14]. He highlights cloud execution, parallel agents, and persistent memory as the main differences [14]. His example: a 28-page Notion messaging audit across five criteria, benchmarked against Coda and Slite, with per-page recommendations in about 20 minutes [14].

Why it matters: This is positioned as a tool for bounded PM work where the output itself matters more than chat [14].

How to apply: Start with a constrained audit or launch-prep task, and use the full guide for six PM use cases, exact prompts, and the prompt spec that Gupta says cuts cost by 60%+ [14].

2) Reusable Claude projects beat blank prompts

The same set of examples shows three high-value project templates for PMs: a strategy critic with devil’s-advocate instructions [2], a customer-insight workflow that turns CSVs into reports [2], and a natural-language data analyst that answers plain-English questions with SQL, charts, and tables [2]. The shared

prompting rule is to give the model exemplars and a clear definition of good work [2].

Why it matters: PM leverage increases when the model has project-specific context instead of starting from zero every time [2].

How to apply: Save one reusable project per recurring workflow and feed each one examples from your own team rather than generic prompting advice [2].

3) Gamma can turn analysis into an executive-ready readout

In the NPS example, Gamma generated an executive summary deck, selected visuals, and structured the presentation automatically from the analysis [2].

Why it matters: It shortens the path from raw insight to stakeholder-ready communication [2].

How to apply: Pair it with a verification step on the underlying analysis so presentation speed does not outrun analysis quality [2].

4) Keep a north star metric review checklist handy

The four-question pressure test from Run the Business is simple enough to reuse as a standing template in roadmap, OKR, or quarterly business reviews [3].

Why it matters: It forces teams to connect metrics to customer value and cross-team alignment, not just target movement [3].

How to apply: Add the four questions as a required review section before approving a new north star [3].

Sources

1. Why PMs who can't build will get left behind | Elena Verna | Product in Practice
2. How I Use AI as a Product Manager
3. Before the Metric Goes South
4. r/ProductManagement comment by u/Kancityshuffle_aw
5. r/ProductManagement post by u/DAnnaTroi
6. r/ProductManagement post by u/Mobile-Influence-371
7. r/ProductManagement comment by u/ThePhychoKid
8. r/ProductManagement comment by u/Daraca
9. r/ProductManagement post by u/the_____overthinker
10. r/ProductManagement comment by u/84tiramisu
11. X post by @lennysan
12. X post by @lennysan
13. X post by @tfadell
14. substack