Scaling Discovery Without Losing Product Judgment

PM Daily Digest

2026-03-13

Scaling Discovery Without Losing Product Judgment

By PM Daily Digest • March 13, 2026

AI is making product judgment, not production, the scarce skill. This brief focuses on how PMs can scale discovery without losing intuition through continuous feedback systems, AI-assisted research, instrumented prototypes, and self-serve analytics.

Big Ideas

1) The scarce skill shifted from building to deciding

Creation is easier; judgment is now the differentiator. Hiten Shah frames the new bottleneck as deciding what should exist, who it is for, what problem it solves, and whether it should be built at all [1]. Rekhi makes the operational PM version: AI has sped up delivery enough that discovery and design can become the constraint, increasing the risk of feature factories if teams stop validating with customers [2].

> The bottleneck moved. [1]

- **Why it matters:** faster output does not automatically create better products.
- **How to apply:** raise the bar on problem selection before prompting or prototyping starts, and require a clear statement of what is being built and why [1].

2) AI should speed discovery, not replace product intuition

Rekhi's core caution is to stay in the loop. PMs still need to read customer feedback, sample interview recordings, and digest research themselves to build intuition and empathy [2]. In his experience, end-to-end workflows that ask

AI to synthesize feedback, invent the feature, and write the spec produce poor results; the better pattern is to let AI surface pain points and summaries, then apply human judgment to the solution [2].

- **Why it matters:** many PM decisions still happen without full information, so intuition remains a real operating asset [2].
- **How to apply:** ask for verbatims alongside summaries, and keep solutioning as a human step [2].

Top 10 AI Hacks for Customer Discovery by Reforge's Sachin Rekhi at Lean Product Meetup (8:51)

3) Discovery is becoming a continuous operating system

The notable change in Rekhi's workflow is not one tool but a connected system: continuous surveys, feedback rivers, AI-generated interview guides, interview synthesis, AI-moderated interviews, functional prototypes with analytics, and natural-language data analysis [2]. He describes moving from quarterly NPS work with a marketing team to continuous collection and weekly automated reporting [2].

- **Why it matters:** PMs can keep pace with faster engineering without waiting on long research or analytics queues.
- **How to apply:** replace one-off studies with a recurring loop: collect, synthesize, validate, instrument, and revisit.

4) Some products may need to design for agent-to-agent use

Aakash Gupta argues that many teams still optimize for a single interaction model - human opens app, human types, AI responds - while the next surface may be agent-to-agent, where a user's assistant contacts the product directly [3]. His suggested PM questions shift to what the product exposes to other agents, what permissions it grants, and where it escalates when an AI cannot answer [3].

- **Why it matters:** onboarding flows, nudges, and empty states assume a human is watching [3].
- **How to apply:** inventory which actions, permissions, and escalation paths would still work if another agent were the caller [3].

Tactical Playbook

1) Build a continuous signal stack

1. **Run surveys continuously.** In Rekhi's example, Claude calculated an overall NPS of 41 from 2,600 responses, showed 58% promoters, visualized monthly trends, and supported deeper segmentation work [2].
2. **Automate the reporting layer.** His agent checks the CSV, runs numerical analysis and verbatim themes, then generates both an HTML report and a Gamma presentation [2].
3. **Add a feedback river.** Tools in this category pull from sources such as App Store, Google Play, G2, Zendesk, cancellation surveys, and NPS, then group feedback into themes with trend lines and counts [2]. In his example, one theme showed 138 complaints about Evernote import issues [2].
4. **Keep the customer voice attached.** Review exact verbatims and use the linked identities or contact records when you need follow-up conversations [2].

Why it matters: this turns customer signal from a quarterly project into a weekly operating rhythm.
Start here: automate one recurring survey report, then add one support or review source next.

2) Upgrade interview work end to end

1. **Seed AI with strong interviewing principles.** Rekhi uses a summary of actionable best practices from *The Mom Test* plus a research brief to generate better interview guides [2].
2. **Transcribe and summarize recordings against a template.** His workflow extracts takeaways, pain points, workflow/tools, feature requests, and direct quotes from each interview [2].
3. **Ask for cross-interview patterns.** The same workflow can cluster themes across a batch and count how often pain points appear across 10

interviews [2].

4. **Use AI-moderated interviews when speed matters.** Rekhi says these tools can run async concept interviews, probe with follow-up questions, and return summarized results by the next morning [2].
5. **Still watch some interviews yourself.** He explicitly keeps sampling recordings to build customer and product intuition [2].

Why it matters: scripting, transcription, and synthesis compress from hours or days into an overnight workflow.
Start here: standardize one interview template before automating anything else.

3) Use prototypes as research instruments

1. **Build a functional prototype, not just a mockup.** Rekhi used Bolt to create a working Ask AI concept [2].
2. **Instrument it like a real product.** He adds in-product surveys, retention tracking, session replay, and heatmaps through analytics tooling such as PostHog [2].
3. **Use synthetic users for fast usability feedback.** In his example, synthetic feedback surfaced a hidden entry point, 5-20 second waits, missing example prompts, and the lack of multi-note querying [2].
4. **Then let real behavior settle design choices.** Heatmaps showed users clicking the Ask AI button in the top right, not the bottom-right floating button [2].
5. **Do not confuse synthetic feedback with product-market fit.** Rekhi says it is useful for usability-style feedback, not for determining product-market fit [2].

Why it matters: this gives teams pre-launch signal on usability and return behavior that mockups cannot provide.
Start here: add one survey question and one heatmap before you send a prototype out.

4) Make analytics self-serve - carefully

1. **Connect AI to your data safely.** Rekhi uses MCP or a database dump with read-only access so the model can inspect schema and query data [2].
2. **Ask questions in plain English.** His workflow reads tables and columns, writes SQL, groups results, and returns charts or recurring dashboards from natural-language prompts [2].
3. **Teach the system with examples.** His advice is to add real question-and-SQL pairs because the model learns patterns from examples well [2, 4].
4. **Document schema quirks explicitly.** A simple instruction such as using `canonical_source` instead of `source` improved accuracy in his example [2, 4].

5. **Audit early, then share.** Rekhi says he audited a few dozen queries to build confidence, and recommends sharing the Claude Project so the whole team benefits from the learned context [4].

Why it matters: PMs can answer more of their own product questions without waiting on a data queue.
Start here: teach one high-value dataset first; do not trust a zero-context setup [4].

Case Studies & Lessons

1) Tesla optimized for the moment of doubt

Tesla's Supercharger spacing is described here as a product decision, not a simple infrastructure rule. The cited explanation is that chargers were placed around where drivers typically hit 15-20% battery - when range anxiety begins and people start doing mental math - rather than at uniform intervals [5]. The deeper move was optimizing for the user's emotional state at a key moment, even though that is harder to put in a dashboard than coverage or utilization [5].

- **Why it worked:** it reduced the moment where users begin to doubt the journey [5].
- **How to apply:** identify the point in your journey where users start calculating, hesitating, or seeking reassurance, and design around that moment first [5].

2) Notejoy's Ask AI concept shows layered discovery in practice

Synthetic users first surfaced usability issues in Rekhi's Ask AI prototype: the entry point was not obvious before a note was selected, response time felt too slow, example prompts were missing, and cross-note querying was absent [2]. After that, prototype instrumentation showed that users preferred the top-right Ask AI button over the bottom-right alternative [2].

- **Why it worked:** different methods answered different questions - synthetic users for early usability friction, real usage for design preference.
- **How to apply:** use simulated feedback to narrow what to test, then use real telemetry to settle decisions [2].

3) LocalMind was interesting, but not frequent enough

Lenny Rachitsky describes LocalMind as an app on top of Foursquare that let people ask questions of users checked into places around the world, such as whether there was a long line at a location [6]. His conclusion was that it solved a real problem, but only occasionally, which made it hard to support as a standalone business [6].

- **Why it matters:** novelty and utility do not guarantee repeated use.

- **How to apply:** pressure-test frequency of need early, not just whether the concept is clever or helpful [6].

Career Corner

1) Strong PM narratives still center on impact, judgment, and alignment

Lenny's shorthand for the role is impact, collaboration, judgment, and alignment, with coordination close behind [6]. He also frames the PM job as delivering business impact by prioritizing and solving the most impactful business problems, while thinking the way a CEO would think about the product's success [6].

> My words are impact, collaboration, judgment, alignment. [6]

How to use it: in interviews and promotion narratives, explain the business problem, the decision you drove, the alignment work, and the outcome.

2) AI hiring signals are moving from prompts to systems

Aakash Gupta says PM interviews now commonly ask how candidates use AI in their workflow, and the answer interviewers want is a system, not a generic tool mention [7]. His examples of strong setups include custom GPTs for PRD drafts, Claude Projects loaded with company design principles, Gemini Gems for competitive analysis, and Claude Code with context; he estimates roughly two hours of setup for 5+ hours of weekly return [7].

How to use it: build one reusable system for writing, one for analysis, and one for recurring tasks - and be ready to explain inputs, outputs, and guardrails [7].

3) Protect the human skills AI can erode

Rekhi's guidance is consistent across research and analytics: use AI to speed production, but keep empathy, pattern recognition, and solution judgment with the PM [2]. He continues to watch interviews and read research himself even after automating much of the process [2].

How to use it: keep a weekly habit of reviewing raw customer material, not just summaries.

4) Classic PM strengths still compound

Lenny points to organization, a high bar for quality, and succinct communication as the traits that carried over from PM work into creator work [6]. He also describes iterating heavily rather than shipping first drafts, sometimes revising a newsletter post dozens of times [6].

How to use it: treat memos, specs, and stakeholder updates the way you treat product flows - refine them until the point is easy to grasp.

Tools & Resources

- **Feedback rivers:** Reforge Insights, Interpret, Craftful, Birdie, Miro Insights, Unwrap, and Productboard for aggregating reviews, support, and survey data into themes and trends [2].
- **Interview synthesis:** NotebookLM and Claude for transcribing recordings, summarizing interviews, and finding patterns across batches [2].
- **AI-moderated interviews:** Reforge, Listen, Outset, and Maze for asynchronous concept testing with dynamic follow-up questions [2].
- **Synthetic user testing:** Reforge and Simile for persona-based usability feedback on prototypes [2].
- **Prototype stack:** Bolt for fast functional prototypes; PostHog for surveys, retention, heatmaps, and session replay [2].
- **Analytics stack:** Claude Projects or Claude Code plus MCP for natural-language SQL and dashboards, especially when paired with example queries and shared instructions [2, 4].
- **Reporting output:** Gamma for auto-generated presentations from recurring analysis workflows [2].
- **Interview prompt seed:** *The Mom Test* as a best-practices source for AI-generated customer interview guides [2].

Sources

1. X post by @hnshah
2. Top 10 AI Hacks for Customer Discovery by Reforge's Sachin Rekhi at Lean Product Meetup
3. substack
4. X post by @sachinrekhi
5. r/ProductManagement post by u/Due-Bet115
6. How I built a 1M+ subscriber newsletter and top 10 tech podcast | Lenny Rachitsky
7. substack