Optimizely compromise: product teams own failures in vendor code too

PM Daily Digest

2026-11-22

Optimizely compromise: product teams own failures in vendor code too

By PM Daily Digest • November 22, 2026

The Optimizely script injection and a run of outages raise questions about vendor risk and failure-state design. Also: simulation as a research tool, the PRD as a decision record, growth loops, and pricing pages written for AI agents.

A compromised A/B testing tool, and what product teams should take from it

Mind the Product's weekly review leads with Optimizely, the experimentation platform many product and growth teams use. Report URI found malicious JavaScript served to visitors of nearly 200 companies, including Zoom, the BBC, CNN and Salesforce. An attacker had put a loader into the field where customers can add their own custom JavaScript. Of 1,611 scripts checked, 188 were infected, across 178 organizations [1]. The payload swapped crypto wallet addresses, skimmed card numbers, and took cookies and screenshots [1]. Optimizely blamed a flaw in the signup flow for its web experimentation product. It said fewer than 5% of customers were affected and that "this was not a credentials breach" [1]. Allowing Optimizely's domain in a content security policy protected nobody, because the bad code came from that domain [1].

The host is careful to say he can't prove AI caused any of the week's failures [1]. His argument is about volume. He cites Veracode: AI writes roughly half of committed code at adopting companies, and about 44% of AI-generated tasks introduced a known vulnerability [1]. He also links this to flatter orgs. Canva-style cuts to layers may also remove the people who caught each other's mistakes [1].

His recommendations: - **Own vendor failures.** Customers blame your brand, not the vendor [1]. Write a one-page plan covering who detects a failure, who

decides what customers are told, and who writes and approves the message. Then run it as a drill [1]. - **Design the failure state.** Pick one critical vendor and write down what users see and can still do during a 24-hour outage. If the answer is "nothing," you've made a product decision by accident [1]. - **Gate prototypes.** "The AI wrote it and the tests pass" is not a definition of done. Someone has to understand the code before it ships [1].

Hiten Shah describes the same review problem from the builder's side. After a day with 3,120 commits from agents, his question became "how am I supposed to review all this work?" [2]. He sums it up: software is getting easier to produce faster than people are getting better at deciding what should exist [3].

Simulation as market research

Scott Belsky writes about startups such as Simile that build simulated populations to test slogans, pricing and other decisions. He says many marketing and price-testing uses aimed at specific demographics are proving more than 90% accurate when projections are compared with outcomes, though the field is early [4]. Product uses he names include whether current customers would pay for an adjacent product, and how a price increase would affect brand sentiment [4]. He expects simulations to replace focus groups and some market research. His caveat: simulations reflect how people think and feel today, so they may miss the edge preferences that later become the center [4]. Shah adds a practice: run the scenarios, make the decision, then study what the simulation got wrong, because "the best ideas might be hiding in the misses" [5].

The PRD as a decision record

Aakash Gupta's updated template assumes you go from an idea to five prototypes and write the PRD for the one that survives. The PRD becomes "a decision record more than a permission document" [6]. What's new for AI features: - a behavior contract of 15–25 example responses, ranging from bad to good to excellent - graduation metrics plus a golden set of offline evals [6] - the rollout's exposure, duration and size, plus kill-switch conditions and an owner - a link to results at the top of the document [6]

In a separate note, he says AI-built apps still need a growth loop. LinkedIn's contact imports, Substack readers who become writers, and Canva designs that act as ads are product decisions made before any code is written [7].

Smaller signals

- **Agent-readable pricing.** A product marketer says developers now ask an agent before opening docs or pricing. In their check, 15 of 16 developer tools served a markdown pricing page, compared with 1 of 12 SaaS products. They ask who should own how agents see the product, and whether anyone tests what ChatGPT or Claude says about their pricing [8].

- **A discovery lesson.** Vedion, an execution-risk diagnostic tool, reached three paid pilots and then shut down. Its founder says leaders already knew where execution was breaking. What they lacked was internal ownership and follow-through, which nobody would pay an outside tool to provide [9].
- **Dev-tool niches.** Commenters say generic AI code reviewers are crowded. Narrow niches like messy industry data or compliance are more open, because the users won't just build the tool themselves [10]. One test: ask 10 engineers what they still do by hand. If three name the same pain and no tool owns it, build that [11].
- **AI PM jobs.** Gupta lists 34 open PM roles at model companies, led by Tencent (7), Google DeepMind (6) and Alibaba (5). Posted base pay reaches $460K at Anthropic and $445K at OpenAI [12].

Sources

1. What went wrong at Microsoft, Workday and the FAA?
2. X post by @hnshah
3. X post by @hnshah
4. X article by @scottbelsky
5. X post by @hnshah
6. substack
7. substack
8. r/ProductMarketing post by u/Joke-Just
9. r/startups post by u/Full_Satisfaction125
10. r/startups comment by u/snap-dev
11. r/startups comment by u/techtuber101
12. substack