

How AI Products Are Actually Built at US Big Tech

Approaches, mistakes, and the new role of the PM

WHERE WE STARTED

The old PM playbook

Built for deterministic software — and it worked well for two decades.

1

Write the PRD

Requirements are fixed before a line of code is written.

2

Engineering estimates

Scope is broken into tickets with a known definition of done.

3

Ship & measure adoption

Success is a dashboard metric, checked weeks after launch.

4

Plan the next quarter

The roadmap is a static document, revisited every 90 days.

What AI actually changes

No single correct output

The same prompt can produce five reasonable answers. “Correct” becomes a distribution, not a spec.

The model moves under you

Vendors ship new model versions monthly. Behavior your product depends on can shift without your team touching code.

Quality lives in the eval, not the doc

A PRD can’t describe every edge case a model will hit. The evaluation suite becomes the real source of truth.

Three recurring mistakes

Patterns that show up across teams and companies, not isolated incidents.

01

Demo-driven development

A polished five-minute demo ships the project. Real-world variance across the full input space is discovered by users, not QA.

02

Chasing the model, not the user

A flagship model launches, and the roadmap reorganizes around it — whether or not it solves a user problem.

03

Prompts as throwaway code

No version control, no tests, no owner. Product logic that decides what users see is treated as disposable text.

MISTAKE 01

Demo quality $\neq$ eval quality

What it looks like

- A curated set of prompts always looks great in review.
- Leadership approves launch based on that handful of examples.
- Real traffic hits the 20% of inputs nobody tested.

The fix

- Build an eval set from real production traffic, not hand-picked prompts.
- Track a quality score over the full distribution, not a demo pass/fail.
- Launch gates require the eval score, not a live walkthrough.

The new PM skillset

Evals as the new PRD

Defining what “good” means, in a form a model can be measured against — before a designer or engineer starts work.

Fluency in reading model behavior

Comfortable enough with prompts, eval dashboards, and failure clusters to have a real conversation with research.

Translator between three functions

Sitting between research (what the model can do), safety (what it should do), and engineering (what ships reliably).

Iterating in days, not quarters

Roadmaps built as living hypotheses, revised as fast as the underlying model itself changes.

From roadmap owner to system trainer

BEFORE

Owns the spec

Plans in quarters

Reviews a demo

Ships a feature

Success = adoption metric

NOW

Owns the evaluation criteria

Plans in sprints, revisits weekly

Reviews an eval report

Ships a behavior, tuned over time

Success = eval score + adoption

Three habits for any PM

Works at a five-person startup or a Big Tech org — the scale changes, the habits don't.

1

Write eval criteria before the spec

Decide how you'll measure "good" before anyone designs or builds.

2

Version prompts like code

Review, test, and roll back prompts the same way you would a backend change.

3

Decouple the roadmap from release cycles

Let user problems set priorities — treat new model versions as an input, not a trigger.

Thank you

How AI Products Are Actually Built at US Big Tech

Andrei · Raptors Conference Proceedings · raptors.dev