Bessemer
Venture Partners

Secrets of an AI-pilled engineering team

tl;dr: The fastest engineering teams in 2026 aren't choosing between speed and quality—they're building the operating infrastructure to sustain both, while actively guarding against the comprehension debt that AI-speed adoption creates.

#1 — Split shipping speed from release risk.

The fastest teams decouple these stages into separate decisions. Ramp ships to an early access tier of 5,000+ opt-in businesses whenever ready. GA is a separate, evidence-gated decision requiring a 7-point checklist covering KPIs, customer feedback, and a rollout plan. Speed and quality stop being a tradeoff the moment you have two release tiers.

#2 — Standardize the infrastructure layer, not the tools.

Shopify doesn't mandate a single AI coding tool, but rather built an internal LLM proxy that routes all AI requests through one centralized gateway. This gives leadership cost control and usage analytics without forcing engineers into a single workflow. In a domain evolving this fast, infrastructure standardization is what enables experimentation without chaos.

#3 — Agentic development changes what engineering leadership means.

Engineers are orchestrators of multiple AI agents running in parallel, not individual contributors writing code. The question for every engineering leader in 2026 isn't whether your team can scale, but rather if you can lead through a transition of what the next era of SWE looks like.

#4 — Comprehension debt is the hidden tax on AI-speed adoption.

As AI generates more code faster, engineers can gradually lose their understanding of the systems they build. The failure mode is invisible until it isn't. For example, reversion rates look fine, PRs ship, then something breaks and no one can find the root cause. Leaders are measuring comprehension as intentionally as you measure productivity.

#5 — Hire for or cultivate technical leadership that helps grow the company into its next stage.

At seed you need someone who ships, at 10 engineers you need someone who has made the architectural one-way-door decisions before, and at 20+ you need to honestly diagnose whether your biggest challenges are people problems or product problems. Each requires a different profile, and leadership needs to assess whether to grow that leadership from within or welcome on new talent without losing instituional knowledge.

#6 — Abdicate the toil. Never abdicate the thinking.

Engineers must understand systems two to three layers below where they're actively working. Weekly demos can be a signal as they surface whether teams understand what they're building, not just whether they're building it faster. Just like the best F1 drivers understand the engine, so should your engineers.