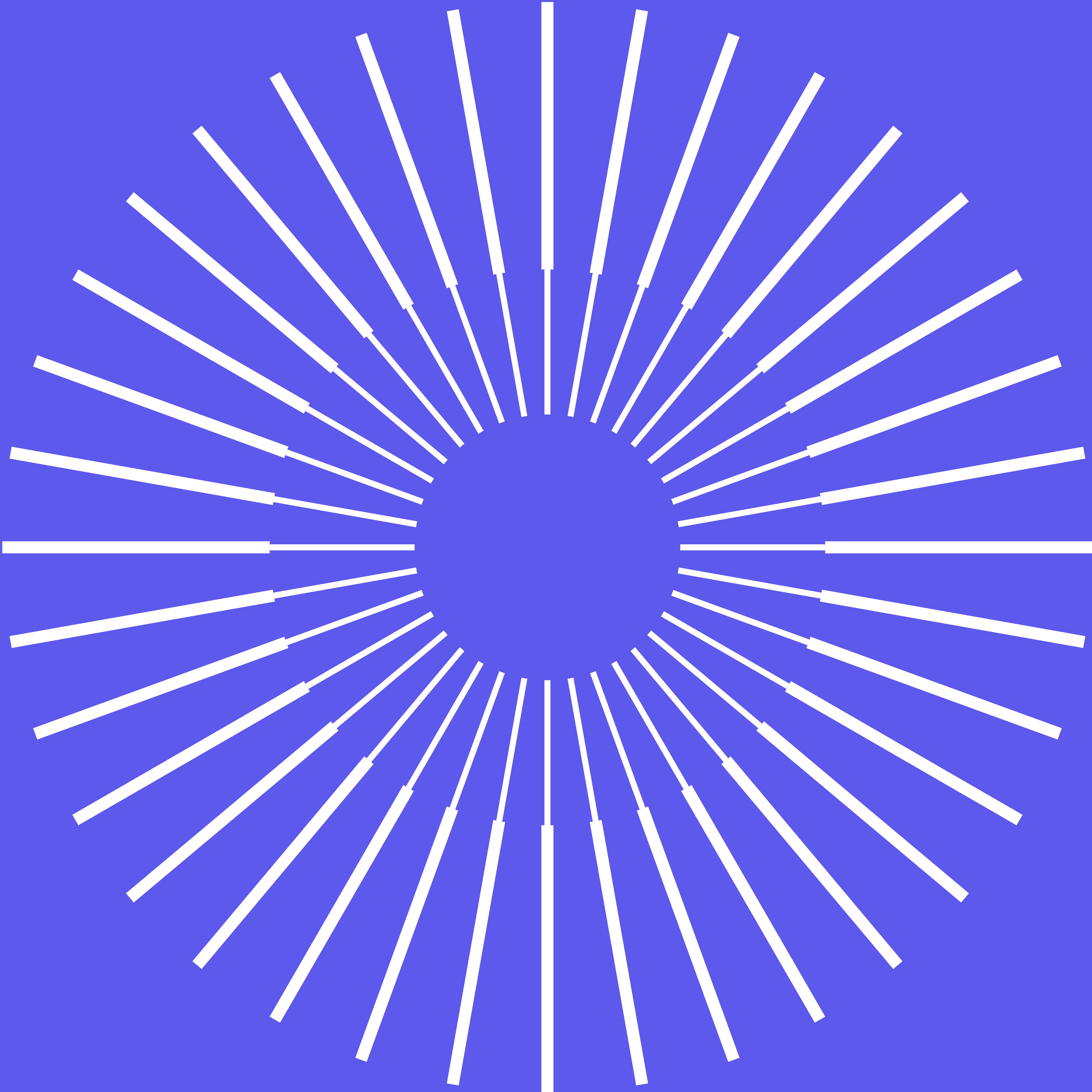




2026 DX Annual
Insights Report

REPORT



2026 DX Annual Insights Report

In April 2026, senior engineering leaders from Airbnb, Microsoft, Uber, Intercom, and others gathered at the inaugural DX Annual conference to compare notes on software delivery in the AI era. This report highlights the key insights and recurring themes that surfaced.

Alongside each insight, the DX Research team—whose peer-reviewed work and DX Core 4 framework shape how the industry measures developer productivity—has added commentary on where the conversation has gone since April.

For more on DX Annual dates, lineups, and registration, go to dxannual.com

The bottleneck of software delivery is no longer writing code

Microsoft's research shows engineers spend only ~14% of their time writing code, so coding was never really the bottleneck in the first place. The other 86% (code review, testing, documentation, story refinement, deployment) remains largely untouched by AI. That's where more meaningful AI gains can be found.

Vanguard found engineers are 30% faster at writing code with AI, but end-to-end cycle time hadn't moved. PMs were still taking 3–5 days to write stories; QA was still running manual tests at sprint end. Until AI is applied across the full lifecycle (which they're making some headway on), faster coding just means developers wait longer for the next bottleneck.

Also at Vanguard, agent-driven implementation took 2 days, but design review added 4, API onboarding 3, security review 5, and deployment approval 2. **The fastest path to ROI in this case isn't a better coding model, but rather eliminating the three-day security review queue.**

Twilio found that top-quartile AI developers took a 14-point hit in perceived review turnaround, even though actual merge time decreased by hours. A four-hour review wait erased the throughput gain from ten-minute code generation.

If code review is now your biggest problem, it was always your biggest problem. AI just made it visible. The same is true of meeting-heavy cultures, coordination overhead, and lack of flow. **AI is an amplifier, not a cause.**

AI gains live in the 86%, not the 14% spent coding

From the DX Research Team

This insight still holds, but the story is becoming more nuanced. New evidence continues to show that AI's largest gains occur during implementation, with much smaller gains making it through review and delivery. Encouragingly, we're now beginning to see improvements in deployment frequency, hinting that organizations are starting to adapt their delivery systems alongside AI adoption. Code review and verification, however, remain among the most persistent sources of friction.

AI token spend is top of mind and organizations are measuring it differently

One company (300-400 engineers, tech) shared their token spend reached \$128,000/week, expecting this to continue climbing to \$150,000/week. They currently operate with "infinite spend" and no caps, **believing the opportunity cost of slowing down adoption is greater than the cost of the bill**, resulting in some developers individually spending up to \$20,000/month on tokens. Despite high overall bills, this company also noted that their cost per pull request is actually decreasing as throughput increases, justifying the investment.

1Password stated that the "AI token bill is really now your new cloud bill" arguing it must be managed with the same rigor as an AWS bill, and questioned why high-powered models are used for simple tasks when smaller, cheaper models would suffice. To lower costs, they suggest organizations forward-project consumption to negotiate better "token per token" rates with model providers, similar to an AWS Enterprise Discount Program (EDP).

Etsy cautioned against top-down mandates that might incentivize "token maxing," where employees generate unnecessary activity just to hit usage metrics.

Microsoft shared that leaders have a responsibility to create "headroom to learn", which includes spending tokens on experimentation that may not create immediate value because the fastest learners will have a competitive advantage. (They also caveated with the understanding that Microsoft is in a unique position here.)

The AI bill is becoming the new cloud bill

From the DX Research Team

The industry has moved from "use AI everywhere" to "use the right AI for the right job." As token spend has accelerated, organizations have become more disciplined about model selection, deterministic alternatives, and measuring return on investment instead of simply maximizing AI usage.

Structured training and skill quality matter more than tool choice

Indeed put all ~2,000 engineers through a structured, tool-agnostic AI course. Those who completed it saw a 36% reduction in coding time. The big differentiator was whether engineers knew how to use their tools effectively.

Under time pressure, **Microsoft** observed that most engineers revert to **single-threaded, pairing-style workflows** rather than orchestrating teams of agents. The behavioral shift hasn't caught up to the tooling shift.

The competitive advantage is AI fluency, not AI tools

From the DX Research Team

This insight has only become stronger. The organizations seeing the greatest returns aren't necessarily those with the newest AI models, but those investing in AI fluency through structured training, shared practices, and better ways of working. As agentic workflows mature, the limiting factor increasingly appears to be human orchestration skills rather than model capability.

Some leaders are adopting more outcome-based metrics

Uber is evolving their measurement to include feature velocity, which captures value delivered regardless of whether a human or agent wrote the code.

Dropbox shared that they adopted feature velocity as their primary measure of engineering effectiveness for the same reason: **it ties directly to outcomes, not activity.**

Organizations are measuring AI through three areas, **covering cost management** (spend by team and project), **engineering effectiveness** (how well AI is being used), and **value impact** (feature velocity, quality indicators).

The future of engineering metrics is outcomes, not activity

From the DX Research Team

This trend has only accelerated. As AI changes how software is produced, organizations are increasingly anchoring measurement on outcomes like quality and business impact, while treating AI adoption, token usage, and coding activity as diagnostic signals rather than success metrics. AI changes how work happens; it doesn't change what engineering organizations are ultimately trying to achieve.

Productivity gains require daily, sustained AI usage

Airbnb segmented engineers by how many hours per day they spent using AI tools. The finding: **engineers who used AI 4+ hours per day more than doubled their output** versus pre-AI baselines. Engineers who used it occasionally saw only modest improvement. The takeaway for leaders is that the conditions for deep, daily usage matter as much as which tool you buy. Org-wide, Airbnb is now seeing 65% higher throughput and 59% AI-authored code with no mandate in place.

Intercom also shows what sustained daily usage looks like at scale: 95.9% of PRs authored by Claude, **throughput doubled in 9 months**, and defect backlog down over 50%. These gains came with strict quality standards as well.

AI impact comes from daily habits, not daily prompts

From the DX Research Team

What we've learned is that organizations seeing the greatest AI gains are not simply encouraging developers to spend more time with AI, they are systematically removing the barriers that prevent sustained adoption. Trust, organizational support, workflow integration, and realistic expectations all appear to be stronger predictors of long-term AI impact than tool deployment

PMs and designers are shipping more code, and organizations are adapting for it

Airbnb has 2× as many AI users as it has developers. PMs, data scientists, and designers are using AI coding tools daily (even their finance team independently onboarded themselves to VS Code to run AI workflows without engineering involvement).

Similarly, Mercari saw a 45.7% drop in accounting tasks and a 60% reduction in help desk workload from AI tools built and operated by non-engineering teams.

Regarding “coding not being the main bottleneck,” **Vanguard is responding to this structurally**. After finding that faster coding didn't meaningfully move end-to-end cycle time, they began embedding AI tools across PMs, designers, and QA—treating the full team, not just engineers, as the unit of productivity improvement.

AI is expanding who gets to build software

From the DX Research Team

AI coding capabilities are increasingly being adopted by product managers, designers, analysts, and other technical roles, expanding who can contribute directly to software delivery. At the same time, organizations are recognizing that accelerating implementation alone rarely transforms delivery speed. The greatest gains will likely come from treating product development as an end-to-end system, embedding AI across planning, design, engineering, testing, and delivery rather than focusing exclusively on software engineers.



Engineering intelligence

For more on DX Annual dates, lineups, and registration, go to dxannual.com

Learn more at getdx.com Copyright © 2026