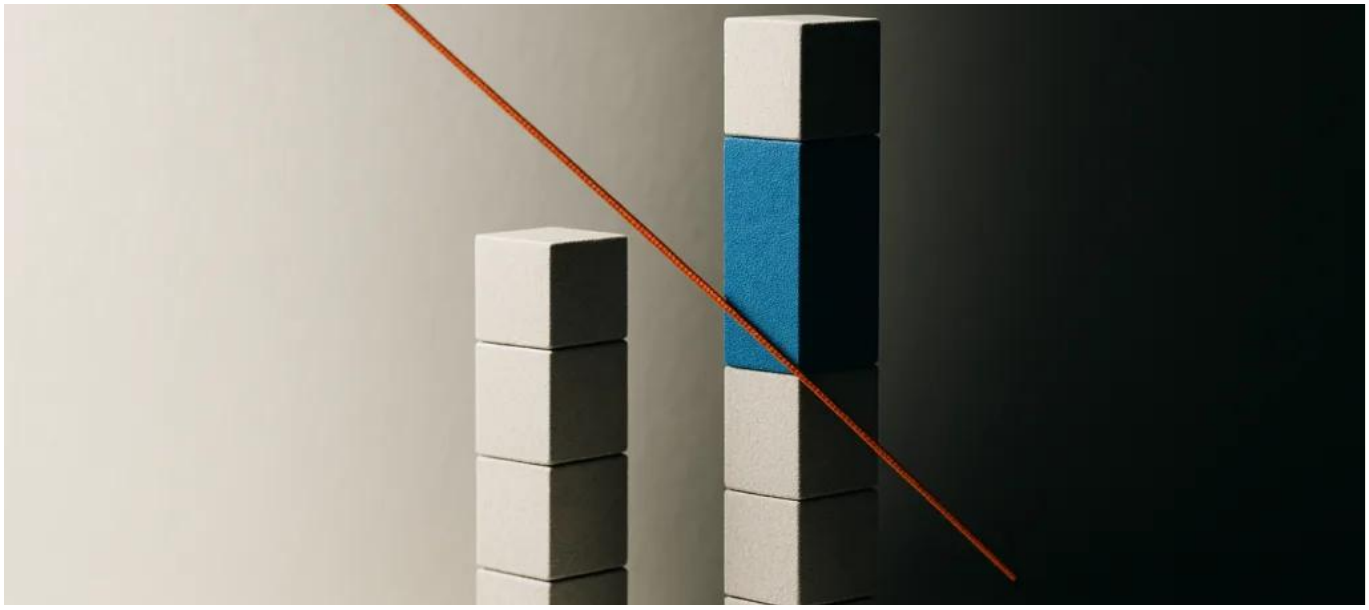


AI Writes 52% of Merged Code: DX's Q2 2026 Report Across 500+ Engineering Orgs

"AI-authored code has nearly doubled, but so has PR size." That is DX's own headline on its Q2 2026 findings, and the second half of that sentence is the part most people will skip.

THE SHORT VERSION

DX's preliminary Q2 2026 data, published August 2026 across 500+ engineering organizations, puts AI-authored merged code at 51.9% on average (median around 50%), up from 34% in Q1 2026 and 24% in 2025. In the same four-quarter window, change confidence fell 6.1%, median PR size rose from 44 to 72 lines, and the Developer Experience Index slipped from 67 to 65.

The figure that reframes things is the pairing, not the 52%. More than half of merged code is now machine-authored, and developers trust that code less than they did a year ago. Volume went up, confidence went down, and both facts come out of the same dataset in the same quarter.

51.9%
of merged code AI-authored, average
DX, State of AI Impact in Engineering Q2 2026 (median ~50%)

-6.1%
change confidence over four quarters
DX Q2 2026 report

+64%
median PR size, 44 to 72 lines
DX, July 2025 to June 2026

~28x
median quarterly AI spend, Tech sector
DX Q2 2026: ~$1.5K to ~$44K in one year

Three quarters of trend data: 24%, then 34%, then 52.7%. The Q1-to-Q2 move alone is a relative jump of roughly 53 to 55%. Adoption is effectively saturated at the individual level, with DX reporting 95% adoption measured through telemetry, which it describes as approaching 100%. The remaining growth comes from the same developers shipping more generated code, not from new developers switching it on.

Throughput rose 37% while PRs grew 64% larger

Median weekly TrueThroughput rose 37%, from 1.42 to 1.94 PRs per engineer per week over four quarters. Self-reported time savings went from 3.9 to 6.2 hours per developer per week, up about 59%, with heavy users reporting over 6 hours. Those are real numbers and they move in the direction the pitch deck promised.

The cost shows up somewhere else. PRs are 64% larger. DX attributes part of the Developer Experience Index drop from 67 to 65 to worsening review turnaround, which is the mechanically obvious consequence of pushing 72-line changes through a review process built for 44-line ones. Reviewers are the constraint now, and nothing in the dataset suggests they got 64% more capacity.

Then the strangest pair in the report: code maintainability improved 3.8% while change confidence fell 6.1%. Code got easier to read and harder to trust. My read is that these measure different things and the gap is diagnostic rather than contradictory. Maintainability rewards surface properties (naming, structure, consistency) that a model is genuinely good at producing. Change confidence is a judgment about blast radius, and that judgment depends on whether the author understood the system well enough to predict what the change touches. Generated code can be tidy and still come from someone who

never traced the call path.

> ! **How to read these numbers**
> DX labels the Q2 figures preliminary. Change confidence and time savings are self-reported survey measures; the 51.9% code share and 95% adoption come from telemetry. Mixing the two in one narrative is useful but not clean: a 6.1% decline in a sentiment index is not the same class of evidence as a telemetry-derived share of merged lines.

The 28x spend increase that bought a flat innovation ratio

Median quarterly AI spend in the Tech sector went from roughly $1.5K to roughly $44K in a year, nearly 28x. Over the same period the innovation ratio stayed flat at 57 to 58%. Teams are spending dramatically more and allocating the same proportion of effort to new work versus maintenance and operational load.

That is the finding I would put in front of a board before any of the productivity numbers. A 28x cost increase that produces 37% more PRs and no shift in what those PRs are for is a specific kind of result: capacity bought, portfolio unchanged. The spend is real. Whether it purchased leverage depends entirely on whether you believe PRs per engineer per week is the thing you were short on.

WHERE I LAND

My take: the 52% figure is a measurement milestone and a terrible management target. What I would act on is the delta pattern: throughput +37%, PR size +64%, confidence -6.1%, DXI down 2 points. Output grew slower than the size of the units flowing through review, and trust fell. That is a review-capacity problem dressed up as an AI-productivity story, and it is fixable with process changes that cost far less than $44K a quarter.

One more number worth sitting with: change failure rate volatility widened, with outlier organizations moving as much as plus or minus 3 percentage points against a 4% industry benchmark. A 3-point swing on a 4% baseline is enormous. The median organization looks stable while the tails pull hard in both directions. Some teams got meaningfully better at not breaking production, others got meaningfully worse. The averages in this report hide

that, and if you are benchmarking yourself against 51.9% or against the 4% failure rate, you are comparing yourself to a midpoint between two divergent groups.

I cannot tell from this data which practices separate the good tail from the bad one. DX does not report that split. My working hypothesis, and I am labeling it as such: the difference is whether review discipline scaled with generation volume, because that is the one lever the reported metrics all touch. A team that held PR size down while adoption rose would show up as throughput-positive and confidence-neutral. A team that let PR size follow generation volume gets the 72-line median and the review-turnaround drag DX names explicitly.

This pattern is not isolated to DX's dataset. The [analysis of 623 million code changes](https://www.arturmarkus.com/ai-code-quality-by-the-numbers-623-million-changes-refactoring-down-70-duplication-up-81/) found refactoring down 70% and duplication up 81%, which describes the same shape from a different angle: more code arriving, less consolidation happening to it. The [METR slowdown study](https://www.arturmarkus.com/the-metr-19-slowdown-was-never-about-ai-being-slow-it-was-about-the-39-point-gap-between-feeling-and-stopwatch/) documented a 39-point gap between how fast developers felt and how fast they were. DX's 6.2 self-reported hours saved per week sits in exactly that category of measure. I would not discard it, and I would not plan capacity on it either.

Cap PR size, pair the spend curve with the innovation ratio, benchmark on the median

Three implications, each tied to a figure in the report.

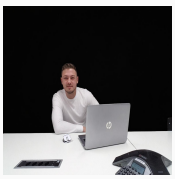
If your PR size has drifted upward the way the median did (44 to 72 lines), that is the first thing to fix, because DX names review turnaround as a contributor to the DXI decline. Caps on PR size are unpopular and they work. The alternative is paying reviewers to context-switch across 64% more diff per unit of judgment.

If you are reporting AI ROI internally, report the spend curve next to the innovation ratio. $1.5K to $44K against a flat 57 to 58% is the honest framing. Anyone presenting the 37% throughput gain without the cost multiplier is presenting half a number.

If you are benchmarking, use the median (~50%) rather than the average (51.9%), and treat the 4% change failure benchmark as a distribution rather than a line. The plus or minus 3 point outlier range is the more useful figure for a risk conversation, because it tells you how far organizations have actually moved in both directions within a single year.

The full report and the August 14 podcast readout are worth reading directly: [DX's Q2 2026 release](https://getdx.com/news/dx-releases-q2-2026-state-of-ai-impact-in-engineering-report/), the [benchmarks readout](https://getdx.com/podcast/ai-in-engineering-q2-2026-benchmarks-research-readout/) with Brian Houck and Justin Reock, and the [blog writeup on PR size](https://getdx.com/blog/ai-authored-code-has-nearly-doubled/) that pairs the two headline movements. The [telemetry methodology for the AI code percentage metric](https://docs.getdx.com/reports/ai-code-percentage.md) is documented separately, which matters if you plan to compute your own number and compare.

Half your merged code is now written by a model, your engineers ship 37% more PRs, and they are 6.1% less sure those PRs are safe. All three are true at once, from one dataset, in one quarter.



WRITTEN BY

Artur Markus

Software engineer in Basel building AI infrastructure and agentic systems, with a background in pharmaceutical automation, GxP compliance, and validation. I help teams turn AI from a chatbot into a reliable operational layer.

Book a meeting →