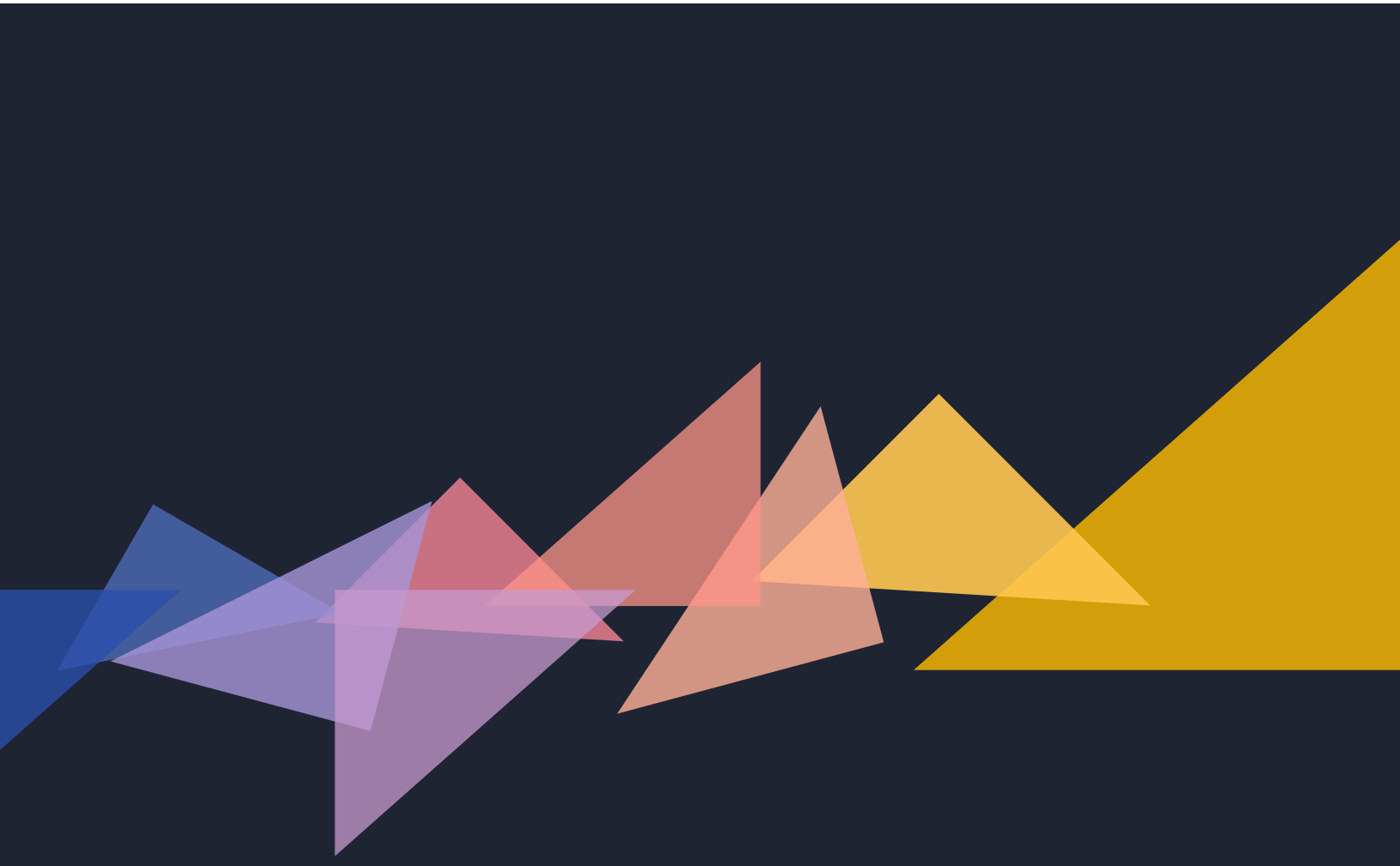
Quotient

QUOTIENT · AI ROI METHODOLOGY

Measuring the ROI of AI-Assisted Software Development

A research-driven methodology for measuring and communicating the value of AI in engineering organizations.



Methodology Whitepaper · Aug 2026 · getquotient.com

Contents

Executive Summary 3

Why AI ROI Became an Urgent Question to Answer 4

What "ROI" Actually Means 6

The Calculations 7

Understanding Weighted Throughput and Value Created 8

Why Throughput Alone Is an Incomplete Picture 11

A Research-Backed, Defensible Answer to ROI 12

Considerations and Limitations 14

Conclusion 16

Appendix: Research Citations 17

Executive Summary

The cost of AI in engineering has risen dramatically, driven by deeper reliance on AI-assisted development, agents completing more work end to end, and pricing that bills for consumption rather than seats. Gartner projects that spending on AI coding agents will surpass the average developer's salary by 2028.

Boards have already reacted. In a Dataiku/Harris Poll survey of 600 enterprise CIOs, 98% reported that board pressure to demonstrate measurable AI ROI has increased since 2024, and 71% said their AI budget would likely be cut or frozen if targets were not met by mid-2026.

Most engineering organizations cannot answer the question, because their instruments count inputs: tokens consumed, suggestions accepted, lines of AI-written code, percentage of developers active in a tool. Those numbers describe usage. None of them show value.

This brief describes the methodology behind the Quotient AI ROI Report: how to translate a measured change in engineering output into dollars, what we hold that figure against to keep it honest, and which alternatives we tested and rejected. It draws on published research on AI-assisted development.

The result is a directional, defensible estimate built from visible assumptions, paired with guardrails on speed, quality, and developer experience so a rising number cannot conceal a degrading system.

Why AI ROI Became an Urgent Question to Answer

AI tooling was once inexpensive enough that most organizations adopted it without a formal business case. Three shifts changed the urgency around investment evaluation.

Pricing moved from seats to consumption

A flat per-developer license was easy to predict, but recent changes in AI costs now attribute spend to usage. Because of this, the cost of AI tooling can dramatically change based on factors such as usage and model selection. Gartner found roughly a quarter of technology leaders already spending $200–$500 per developer per month on tokens, and about 6% above $2,000.

Unit prices are not the driver. Per-token inference costs have fallen sharply (DORA cites a 280-fold drop between late 2022 and late 2024), but the scope of usage has increased exponentially: an agent that plans, edits, tests, and retries pushes far more tokens through a task than a suggestion engine does.

Costs grew exponentially and could not be modeled

Uber reportedly exhausted its 2026 AI budget in four months. Worse, their COO said he could not tie that usage to attributable value when reporting the cost increase.

The pattern is widespread. In a 2026 survey of 396 organizations, 95% had assigned a formal AI budget but only 11% could forecast AI spend within ±10%, and 62% said a cost surprise had altered a business decision in the past year.

AI spend is no longer small, either. BCG puts planned AI spend at 1.7% of revenue for 2026, roughly double the prior year, and most organizations are still investing ahead of visible returns. At that scale a forecast miss is a material variance, not a rounding error. Finance notices, and the question shifts from what AI costs to what it returns.

WHY AI ROI BECAME AN URGENT QUESTION TO ANSWER

The Pressure Shifted to Predicting and Validating Value

Spend that cannot be forecast has to be justified twice: once to get approved, and again to stay funded. Leaders are now asked to project what AI will return before the budget is granted, then show that the return arrived. Both halves need measurement most organizations do not have.

> **98%** of 600 enterprise CIOs report increased board pressure to show measurable AI ROI since 2024; **71%** expect budget cuts or freezes if targets are missed (Dataiku/Harris Poll, 2026).

> **87%** of 260 finance executives say they need to tie AI spend to business outcomes within the year. Only **22%** can do it today (CloudZero, 2026).

Measure	Share of CIOs
Report that board pressure to show measurable AI ROI has increased since 2024	**98%**
Say their AI budget will likely be cut or frozen if targets aren't met by mid-2026	**71%**
Can link half or more of their AI initiatives to measurable cost savings or revenue	**<40%**

Figure 1: Dataiku/Harris Poll survey of 600 enterprise CIOs worldwide, 2026.

Nearly every CIO is being asked the question, and most cannot answer it for the majority of their portfolio. Spending that cannot be defended leaves two bad outcomes: keep funding a program nobody can justify, or cut one that was working because nobody could show it.

KEY INSIGHT

Pressure is increasing to show and justify a measurable return on AI spend. That means a measure whose inputs trace back to system data, whose assumptions are visible and adjustable, and whose limitations are stated rather than discovered later.

What "ROI" Actually Means

ROI refers to return on investment. This section defines each term as it applies to AI-assisted engineering work.

Investment

An organization's AI spend: tool licenses, seats, token and compute cost, taken from billing records.

Value

The additional engineering output a team produces at the same cost, which is capacity the business would otherwise have to hire for. We measure it as complexity-weighted throughput, held against speed and quality. It is a proxy: downstream business value depends on factors outside the engineering system. *(See "Considerations and Limitations" for why we use output rather than extending the estimate into revenue and retention.)*

Return

The extra output translated into dollars and compared against the spend that helped produce it.

> In short: **investment** is total AI spend; **value** is the added engineering output AI helped unlock; **return** is that value expressed in dollars, relative to the spend.

The Calculations

The report resolves to two equations. The first turns a productivity gain into dollars. The second sizes it against spend.

Value Created = **total gain in weighted throughput** × **engineering org cost**

We measure the value created by AI through the additional engineering capacity it gives a team. If it delivers *X%* more real engineering work than its baseline, we value that increase at *X%* of the team's total engineering cost for the same period.

Net ROI = (**Value Created** − **AI Spend**) ÷ **AI Spend**

First subtract the AI Spend from the Value Created to get the Net Gain, then divide by the spend used to create that value to get Net ROI. Multiply by 100 to express as a percentage.

We also report a Gross Multiple, Value ÷ Spend, for audiences who prefer it. The two are the same measurement one step apart: a 1.25× Gross Multiple is a 25% Net Return. The multiple is harder to misread in a board deck, while the percentage reads more naturally in a finance review.

WORKED EXAMPLE — ILLUSTRATIVE

A 45-engineer org, measured over its AI adoption window

Weighted throughput vs. baseline	**+15%**
Organizational cost	**$10.0M / yr**
Value Created = 15% × $10.0M	**$1.5M / yr**
AI spend (annualized)	**$1.2M / yr**

Gross Multiple = $1.5M ÷ $1.2M

1.25×

Net ROI = ($1.5M − $1.2M) ÷ $1.2M

25%

Understanding Weighted Throughput and Value Created

Throughput drives the dollar figure, so how it is counted determines the result. Raw counts of PRs, commits, or lines are unreliable for this purpose, because AI inflates all three. For example, the same work split into smaller pieces, or written with more lines, can be incorrectly registered as more output.

Three adjustments produce a measure that reflects work actually delivered.

1 · Normalize per engineer

Throughput can change drastically based on headcount changes alone. That is why we recommend looking at throughput per engineer per week rather than as a team-wide total.

2 · Weight by complexity

Each unit of work is scored so a substantial change counts for more than a one-line tweak and volume alone cannot move the number. Rather than ranking by percentile, we compare each PR against the average PR in a fixed baseline period, taking three ratios and averaging them:

Files ratio	Comments ratio	LOC ratio
PR files ÷ avg baseline files	PR comments ÷ avg baseline comments	PR LOC ÷ avg baseline LOC

```
PR weight = ( Files ratio + Comments ratio + LOC ratio ) ÷ 3
```

UNDERSTANDING WEIGHTED THROUGHPUT AND VALUE CREATED

3 · Accumulate against a baseline

Measure how much extra throughput accumulated across the full period relative to an early-period baseline. This captures the total gain over the period rather than the rate at a single, arbitrary end point.

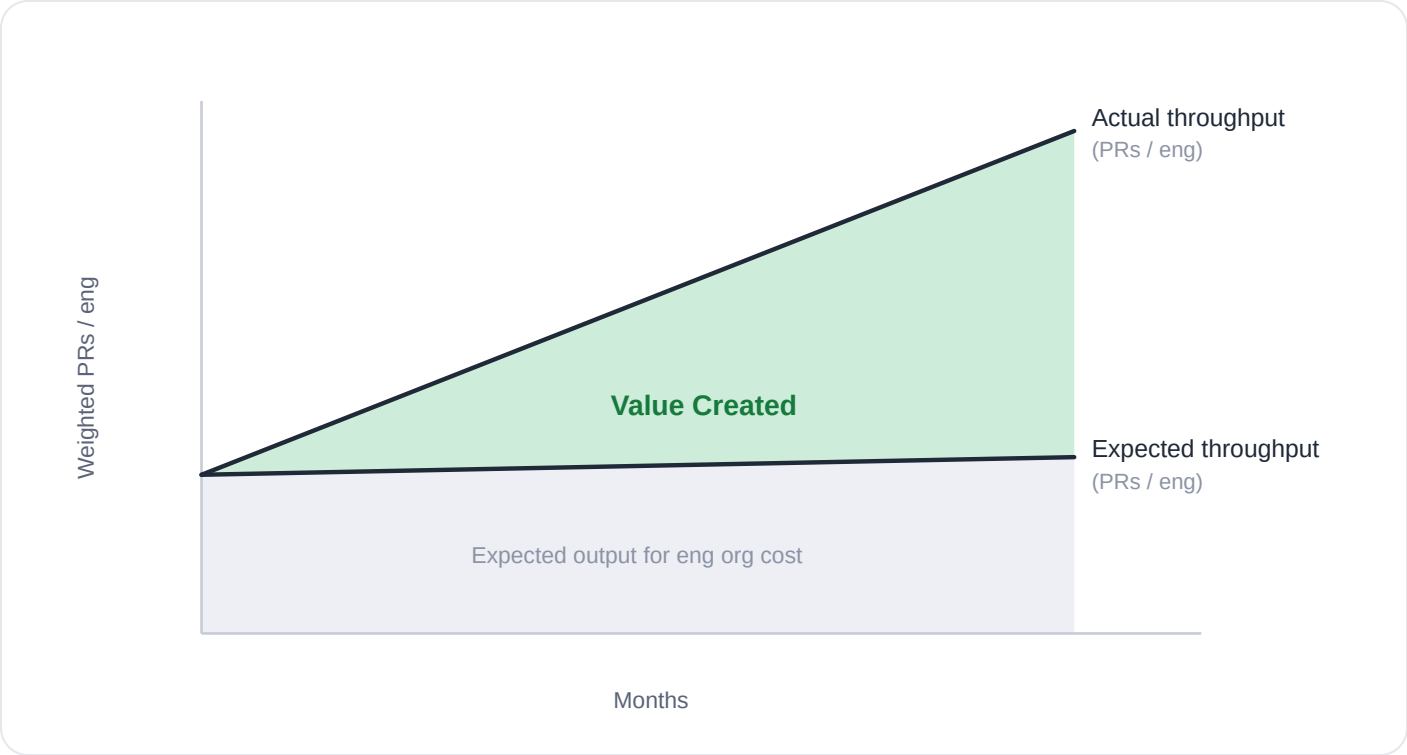


Figure 1: The value AI created is the whole shaded area between actual and expected throughput across every month, not just the gap in the final month. That is why the gain is measured cumulatively against the baseline rather than by comparing two endpoints.

UNDERSTANDING WEIGHTED THROUGHPUT AND VALUE CREATED

Supporting Research

Two findings from recent research explain why throughput is measured from system data and weighted by complexity rather than taken from raw counts or self-report.

RESEARCH: SENTIMENT AND MEASUREMENT CAN DIVERGE

Developer sentiment captures aspects of the work that system data tends to miss, which is why it is a critical input for understanding broader productivity. The evidence suggests it is a weaker basis for sizing hours or dollars saved.

In METR's 2025 randomized controlled trial, developers overestimated AI's effect on their task time by more than 40 percentage points. A 2026 METR survey of 349 technical workers found a related pattern: median self-reported speed gains of 3x against self-reported value gains of 1.4–2x, which METR attributes to task substitution, since AI makes some low-value work cheap enough to do more of it.

Self-report is therefore a difficult basis for a multiplier. That points toward observational measurement from system data against a team's own baseline, with sentiment reported as a check on it.

RESEARCH: THE SIZE OF THE GAIN DEPENDS ON THE KIND OF WORK

Task composition determines how large a gain is available. The 2026 DORA ROI report finds AI delivering a 35–40% productivity gain on simple, greenfield tasks and often 10% or less on complex, legacy brownfield code, a spread of three to four times. A measure that treats all units of work as equivalent absorbs that spread as noise, which is why throughput is weighted by complexity.

Why Throughput Alone Is an Incomplete Picture

Weighted throughput is necessary but not sufficient. It can climb without a team genuinely getting better. Notable examples include starting more at once (more work-in-flight, each item actually slower), sacrificing quality, or working more (i.e. overtime, increasing burnout). Therefore we introduce three measurable guardrails that keep it honest.

Speed Issue cycle time

Elapsed time from a ticket entering "in progress" to being resolved, in business hours, reported as the median to reduce the impact of extreme outliers.

Quality Change failure rate + code quality

This ensures a team cannot win on throughput by shipping at the expense of unstable code.

Developer experience Composite of sentiment signals

Job satisfaction, productivity with AI tools, and time spent on valuable work. The human check that gains are sustainable rather than borrowed against burnout.

Read together, these turn a single number into a defensible picture: the team is producing more, it is genuinely faster, quality held, and the team can sustain the pace.

A Research-Backed, Defensible Answer to ROI

Research consistently shows that AI acts as an amplifier: returns come from the health of the engineering system it operates within, rather than from the tools themselves. Throughput, cycle time, failure rate, and developer experience are direct measures of that system.

Measuring ROI with Quotient

Quotient implements this methodology as a live view of ROI that updates as the underlying data moves, so the figure reflects the current window rather than a point-in-time analysis. The report presents the return alongside the components behind it: AI spend, throughput, speed, quality, and developer experience, each with its own trend against the baseline. Inputs draw on system-of-record data from source control, issue tracking, AI tooling, and developer sentiment signals.

An ROI overview that can be leveraged for an exec or board deck.

Alongside ROI, Quotient measures engineering effectiveness across the delivery lifecycle and surfaces the bottlenecks constraining it. Engineering leaders use Quotient to answer the ROI question with confidence and make improvements across the SDLC to improve AI's value and drive transformation.

A RESEARCH-BACKED, DEFENSIBLE ANSWER TO ROI

The Systems That Drive ROI Calculations

Measuring ROI can quickly become a time-consuming venture involving management of various integrations, data cleaning, and validation. To achieve an accurate estimate, readers can leverage a small, well-understood set of sources.

Source control
One source-control system, for throughput and complexity.

Issue tracking
Used to calculate the speed measure, issue cycle time.

AI spend
Seat, and token/compute cost. The denominator of the ROI, knowable from billing.

Developer experience data (via surveys)
A monthly pulse feeds the quality and developer-experience guardrails. Code quality is often a leading indicator before bugs surface downstream.

Four properties make the resulting figure defensible in a budget review:

- → **It requires no authorship data.** There is no need to establish who, or what, typed which line.
- → **It uses a unit leadership reports on.** Capacity in dollars, set against spend.
- → **Volume alone cannot move it.** Complexity weighting means splitting the same work into more pieces changes nothing.
- → **It shows its derivation.** Every assumption is visible and adjustable, so the figure can be walked through line by line in a board meeting.

Throughput is reported alongside its guardrails, so a gain achieved at the cost of slower cycle times, declining quality, or a worsening developer experience remains visible in the results.

Considerations and Limitations

Developing this methodology required choosing among several published approaches to measuring the value of AI-assisted development. This section documents what was considered, the reasoning behind each decision, and the questions that remain open.

Decisions we made against

Attributing output to AI authorship.

A commonly-used measure we encountered was the share of code written by AI. Several considerations weigh against it: authorship is difficult to verify reliably at scale, it describes an input rather than a value, and measurement built on it raises reasonable privacy concerns among developers. The calculation does not depend on knowing the origin of individual lines.

Survey-based time saved as the primary input.

DORA's calculator starts from estimated net time saved per developer. Perceived gains run consistently positive and couple loosely to system-level outcomes, so self-report serves as a guardrail signal instead.

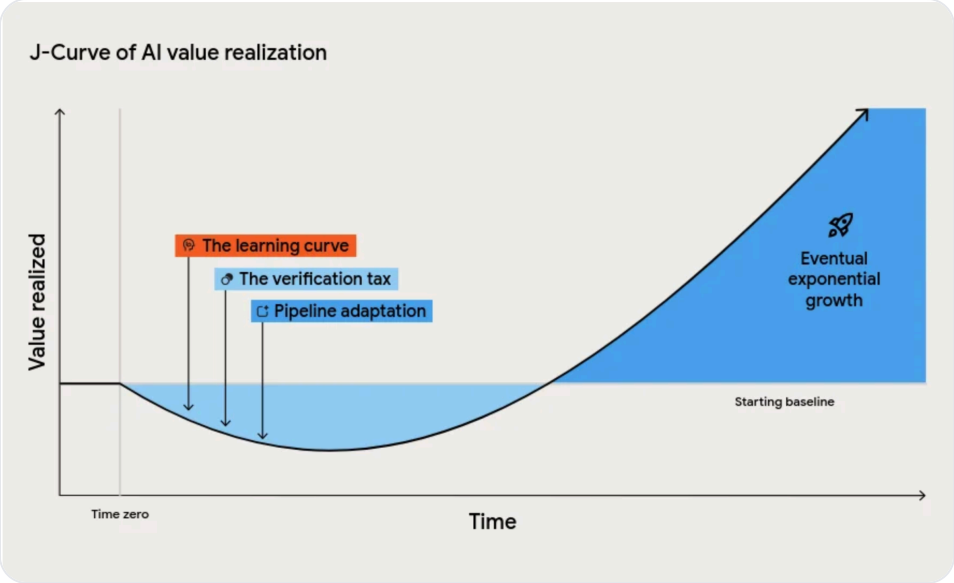
Extending value into revenue.

Some models extend into revenue from accelerated delivery and business growth, acknowledging its levers differ in how firmly they can be established. This methodology stops at delivery, because the chain from shipped feature to recognized revenue runs through variables outside the engineering system.

Limitations of the Methodology

Several constraints bound how precisely the resulting figure can be stated. Each is worth understanding before the number is used in a budget decision.

- **The estimate is directional.** It is assembled from visible, adjustable assumptions, so changing an input changes the result.
- **The relationship is associative.** The throughput gain coincides with the AI adoption period. Establishing causation would require a control group, and an organization measured against its own past has no comparison arm. The 2026 METR study noted that since too many developers now decline to work without AI, control groups are increasingly difficult to measure against.
- **The measurement window affects the figure.** DORA anticipates a J-curve in AI adoption: a temporary dip in productivity while teams learn the tools and downstream processes absorb the new volume, followed by gains as both adjust. A window that includes the dip yields a lower number; one that starts after it yields a higher one.
- **Organizational cost is a composite figure.** Fully-loaded engineering cost must be collected from financial data rather than engineering systems, and organizations assemble it differently.
- **Guardrail signals are not netted out of the figure.** The dollar value derives from throughput alone. Declining quality, slower cycle times, and a worsening developer experience all reduce future capacity, but none of them subtracts from the current estimate.



The J-curve of AI value realization (2026 DORA ROI Report).

Conclusion

AI spend in engineering is increasingly governed like other large capital allocation costs. Most organizations are being asked to account for it without the right framework, which leaves engineering leaders defending a growing budget with data that describe usage rather than outcomes.

Quotient's AI methodology was designed to provide a defensible, trustworthy answer to that question. Each component was chosen against published research on AI-assisted development, tested against the alternatives available, and kept when the evidence supported it. The approaches considered and set aside are documented here, as are the constraints.

This ROI methodology gives an engineering leader a straight answer to an increasingly urgent financial question. With a dollar figure in hand, AI spend can be evaluated like any other investment: weighed against the headcount, tooling, and infrastructure competing for the same budget, defended where the return holds up, and redirected where it has yet to.

Quotient's AI ROI report calculates this number from an organization's own engineering data, and also quantifies how AI adoption is moving delivery outcomes, and which bottlenecks are holding the figure down.

See it at **getquotient.com/ai-roi**

Appendix: Research Citations

The following represent the primary research informing this methodology. Summaries of many of these papers, and additional research reviewed, are available at rdel.substack.com.

AI-ASSISTED DEVELOPMENT AND DELIVERY OUTCOMES

[1] DORA, The ROI of AI-assisted Software Development (2026). https://dora.dev/ai/roi/

[2] DORA, 2025 State of AI-assisted Software Development. https://dora.dev/dora-report-2025

[3] DORA, Software delivery performance metrics. https://dora.dev/guides/dora-metrics

[4] Stanford, Software Engineering Productivity Research (greenfield vs. brownfield gains). https://softwareengineeringproductivity.stanford.edu

[5] METR, Measuring the Impact of Early-2025 AI on Experienced Open Source Developer Productivity. https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/

[6] METR, We Are Changing Our Developer Productivity Experiment Design (February 2026). https://metr.org/blog/2026-02-24-uplift-update/

[7] Becker, J. (METR), Measuring the Self-Reported Impact of Early-2026 AI on Technical Worker Productivity (survey of 349 technical workers, February–April 2026). https://metr.org/blog/2026-05-11-ai-usage-survey/

SPEND, PRICING, AND EXECUTIVE PRESSURE

[8] Gartner forecast on AI coding agent spend, as reported by DevOps.com (2026). https://devops.com/ai-coding-costs-could-exceed-developer-salaries-gartner-warns/

[9] Dataiku / Harris Poll, survey of 600 enterprise CIOs on AI ROI pressure (2026). https://www.dataiku.com/blog/ai-budgets-measurable-proof

[10] CloudZero, survey of 260 finance executives on proving AI ROI (2026). https://www.cloudzero.com/finance-needs-ai-roi-2026-survey-report/

[11] KPMG, Global AI Pulse Survey, Q2 2026 (2,145 senior leaders across 20 countries).

[12] Mavvrik with Benchmarkit, 2026 State of AI Cost Governance Report (396 organizations, April–May 2026). https://www.mavvrik.ai/blog/blog-ai-cost-governance-report-2026/

[13] BCG, planned AI spend as a share of revenue, 2026. https://www.mavvrik.ai/blog/ai-cost-statistics-2026/

[14] Angelo, J., Uber burned through its entire 2026 AI budget in four months, Fortune, May 26, 2026. https://fortune.com/2026/05/26/uber-coo-ai-spending-tokens-claude-code/

MEASUREMENT METHOD

[15] Curti, C. and Williams, C. (Atlassian), A Score for Pull Request Complexity: Its Impact on Cycle Time and How We Reduced It with AI, DPE Summit 2024. https://dpe.org/sessions/caterina-curti-chris-williams/a-score-for-pull-request-complexity-its-impact-on-cycle-time-and-how-we-reduced-it-with-ai/