A RILLET WHITE PAPER FOR FINANCE LEADERS · JUNE 2026

From Tokenmaxxing to ROI: A CFO's Guide to AI Spend

How finance leaders turn runaway AI bills into measurable return.

Rillet

The Morning After

For two years the rule inside most engineering organizations was that more AI is better. Budgets ran open-ended, consumption was the scoreboard, and some companies put usage on internal leaderboards to reward the engineers who burned the most. That era, the one that earned the nickname tokenmaxxing, is over.

Uber burned through its entire 2026 AI budget in four months, then capped spending at $1,500 per employee per tool.[1] Microsoft pulled Claude Code from one of its divisions and routed engineers to its own tooling.[2] Meta's CTO told staff that "token usage alone is not a measure of impact of any kind."[3] Finance teams that never modeled this line item opened the books to a bill nobody planned for. As Ramp's Eric Glyman put it, token spend is "not a clean area of spend."[4]

The sums are not rounding errors. Gartner put worldwide generative-AI spending at $644 billion in 2025, and the average Global 2000 company expects its model bill to climb from roughly $7 million to $11.6 million this year.[5]

This experience isn't unique; we feel it at Rillet too.

The reflex is to clamp down: set a cap, wire up alerts, and call it discipline. That instinct is understandable and mostly wrong. Two things are true at once. The productivity gains from AI are real, and most of the tokens companies buy are not turning into shipped work. The job in front of finance is not to suppress spend. It is to convert spend into return. This paper is about how to do that without strangling the upside.

[1]Uber capped AI spending at $1,500 per employee per tool after exhausting its annual AI budget in four months. TechCrunch, June 2, 2026, citing Bloomberg and The Information.

[2]Microsoft's Experiences and Devices division phased out Claude Code licenses in favor of GitHub Copilot CLI. Quartz, June 11, 2026, citing The Verge and an internal memo from Rajesh Jha.

[3]Meta CTO Andrew Bosworth, internal memo, April 2026, via The Wall Street Journal; reported in Quartz, June 11, 2026.

[4]Eric Glyman, co-CEO of Ramp, on finance teams hit with unplanned AI bills. CNBC, June 26, 2026.

[5]Gartner, Forecast: AI Spending Worldwide (2025–2026), on worldwide generative-AI spending. Global 2000 per-company LLM-spend estimate from 2025–2026 industry research. Figures not independently verified — confirm against the primary reports before publishing.

Why the Bill Exploded

The sticker shock has a cause, and it is not only wider adoption. The work itself got heavier. For two years most AI usage was a person typing a question into a chat box, a cheap and roughly fixed cost. That has given way to agents that retrieve context, call tools, and reason across many steps, re-reading their own growing context before each action. Agentic tasks can consume on the order of a thousand times the tokens of a simple chat exchange,[6] and reasoning models add to it, quietly generating tens of thousands of extra thinking tokens on a single query.

Falling prices make this worse, not better. The unit cost of intelligence keeps dropping, but cheaper tokens invite more use cases and higher volume, so total spend climbs even as each token gets cheaper. It is the Jevons paradox applied to compute.[7] A budget built on last year's prices is already out of date. This is why finance keeps getting surprised: the line item is not only growing, its unit economics are shifting underneath it.

[6]Stanford Digital Economy Lab, on the token intensity of agentic tasks versus chat (the "context snowball"). Not independently verified — confirm against the primary source before publishing.

[7]Menlo Ventures, State of Generative AI in the Enterprise (2025), on rising aggregate GenAI spend alongside falling per-token prices (a Jevons-paradox dynamic). Not independently verified — confirm before publishing.

Caps Are a Blunt Instrument

A spending cap feels like control. It rarely is.

A cap treats every token the same, whether it shipped a feature or re-ran a frontier model on a task a cheaper one could have handled. The price gap is enormous: by one analysis, output tokens from a frontier model cost roughly 83 times those of a small open-weight model.[8] A flat cap cannot see that difference. It just means the engineer running the wrong model for the job hits the ceiling faster, with less to show for it.

Caps also aim at the wrong people. When one engineering organization examined its usage, 91 percent of employees were never hitting their limits in the first place.[9] Lowering the cap mostly generates alerts and friction for people who were not the problem, while the heaviest users route around it with exception requests.

None of this makes caps useless. Uber's separate-cap-per-tool design is meant to force a conversation about cost, not to end AI use.[10] A cap can buy time. On its own, it cannot produce a return. For that you need to know what you are getting, and that turns out to be the hard part.

[8]Model-pricing analysis cited in Quartz, June 11, 2026: frontier output tokens cost roughly 83 times those of a small open-weight model.

[9]Figures shared publicly by an engineering leader at Coinbase on X (91% of employees never hit usage caps). Sourced from social media and not independently verified — confirm before publishing.

[10]Uber's per-tool cap is designed to force a cost conversation, not to eliminate AI use. TechCrunch, June 2, 2026, citing Bloomberg.

The Measurement Gap

Token consumption became a stand-in for productivity because it was easy to count. It is a broken proxy.

Lines of code went up. Output mostly did not. **An NBER working paper tracking more than 100,000 GitHub developers found that coding agents drove a 741 percent increase in lines of code and a 65 percent jump in pull requests, while actual software releases rose only 20 percent.**[11] The bottleneck moved downstream, to the humans who review, test, and ship. The same researchers estimated that even with full upstream automation, output gains would top out near 26 percent until teams fix how work gets reviewed and released.

The pattern holds at the company level. A McKinsey survey found more than 80 percent of organizations were not seeing a tangible effect on earnings from generative AI.[12] MIT's NANDA initiative reported that 95 percent of enterprise AI pilots delivered no measurable financial return, with flawed integration into real workflows as the main culprit.[13] The tools worked for individuals and stalled inside organizations.

Boris Cherny, who built Claude Code at Anthropic, argues the comparison itself is the mistake. Do not benchmark an AI coding tool against a $20 software subscription. Benchmark it against what the work would have cost in engineer-hours.[14] He describes a developer who rewrote an entire codebase from one language to another in six days, a project he estimates would otherwise have taken a year. Measured against a subscription, that spend looks reckless. Measured against a year of engineering salary, it is a bargain.

Both things can be true. The denominator most companies use is wrong, and most of the tokens they buy still are not producing the engineer-year outcomes that justify the bill. The productivity itself is genuine but uneven: controlled studies have shown developers completing tasks far faster with AI, and other rigorous trials have shown experienced engineers running slower on complex work while believing they were faster.[15] Closing the gap between activity and outcome is the actual work.

[11]NBER working paper tracking more than 100,000 GitHub developers, cited in Quartz, June 11, 2026.

[12]McKinsey survey on generative-AI earnings impact, cited in Quartz, June 11, 2026.

[13]MIT NANDA initiative, "GenAI Divide" report, August 2025, cited in Quartz, June 11, 2026.

[14]Boris Cherny, head of Claude Code at Anthropic, speaking at Fortune Brainstorm Tech. Fortune, June 9, 2026.

[15]Productivity evidence is mixed: a controlled GitHub Copilot experiment found developers completed a task 55.8% faster, while a METR randomized trial (early 2025) found a 19% slowdown among experienced open-source developers on complex tasks. Cited in Quartz, June 11, 2026.

What Good Looks Like

The companies getting a return are not the ones spending the least. They are the ones converting spend into output by fixing the infrastructure underneath it. Four moves matter most.

Better defaults. Most employees never choose a model deliberately. They use whatever the tool opens with, so the default quietly sets most of your bill. Changing the default to a cheaper or open-weight model, while letting engineers upgrade for the tasks that warrant it, moves more spend than any cap. When AI startup Lindy switched its default off frontier models to a lower-cost open-weight alternative, CEO Flo Crivello said the cost curve "crash[ed] to the ground" and expects to save millions.[16] The organization that found 91 percent of staff never hit their caps did exactly this rather than lowering the ceiling.

Better routing. A frontier model earns its price for planning and architecture. For routine execution it is often overkill. **Yet roughly 95 percent of enterprise AI usage still runs on frontier models, according to Glean's CEO.**[17] Routing each task to the right model, rather than the most expensive one by habit, is one of the largest untapped savings in most AI budgets.

Better caching. Every time a model runs, it reprocesses a lot of the same material: the same instructions, the same files, the same background context. Caching stores that work and reuses it, so you're not paying full price to reprocess identical inputs on every request. It's mostly an engineering-hygiene fix, and it's one of the largest recurring savings available, because it cuts costs on work the model has already done.

Leaner context and clear visibility. Starting fresh sessions when switching tasks, scoping file context narrowly, and disconnecting unused tools cut waste without cutting work. The goal is not fewer tokens used. There are fewer tokens wasted. And none of it sticks without visibility: engineers and their managers need to see what they spend, broken out by team, the way they see any other operating cost.

The providers are finally making this possible. Anthropic added spend caps and usage analytics at the organization and individual level, OpenAI shipped admin controls and per-employee budget visibility, and GitHub moved to usage-based billing that exposes the cost of each interaction for the first time.[18] The infrastructure to manage this like a real budget now exists. Most companies have not wired it in.

[16]Lindy CEO Flo Crivello, on moving the company's default model to a lower-cost open-weight alternative (DeepSeek); he said the shift sent the cost curve crashing and expects to save millions. CNBC, June 26, 2026.

[17]Arvind Jain, CEO of Glean, on the share of enterprise AI usage still running on frontier models. CNBC, June 26, 2026.

[18]Provider spend controls: Anthropic organization- and user-level caps with analytics; OpenAI admin controls and per-employee budget visibility; GitHub usage-based billing effective June 1, 2026. CNBC, June 26, 2026; Quartz, June 11, 2026.

The CFO's Playbook

This is where finance owns the outcome. Managing AI spend has gone from niche to nearly universal: the share of FinOps practitioners responsible for it jumped from 31 percent in 2024 to 98 percent in 2026.[19] McKinsey's measurement model splits the work cleanly: the CTO owns the technical metrics at the bottom, such as cost per query, and the CFO owns the financial outcomes at the top, such as cost reduction and operating profit.[20] Five practical moves turn that ownership into results.

1. **Change the default, not the ceiling.** Set the organization's default to a cheaper, slower model tier and let employees upgrade manually when the task justifies it. You will move more spend than any cap, with far fewer alerts.
2. **Make spend visible by team.** Work with IT to break out AI costs per department, so every team sees its own bill and the heaviest users know they are being watched, not blocked.
3. **Run a weekly ROI ritual.** Have each person building with AI show what they shipped, not how many tokens they burned. The cadence does the governing.
4. **Name what AI replaced.** Define the specific tools, hours, or headcount each AI investment stands in for. That is the denominator that makes ROI legible to the board.
5. **Route, don't ration.** Match each task to the right model, and reuse cached context, before reaching for a cap. Rationing a frontier model on every task wastes money; routing to the cheapest model that does the job well saves it without slowing anyone down.

The payoff for getting this right is large. BCG found that the companies it classified as most advanced on AI achieved five times the revenue gains and three times the cost reductions of their peers, and the differentiator was not spending more. It was scaling initiatives with clear outcomes and systematic measurement.[21]

[19]FinOps Foundation, State of FinOps 2026, on the share of practitioners responsible for AI spend rising from 31% (2024) to 98% (2026). Not independently verified — confirm against the primary report before publishing.

[20]McKinsey five-layer AI measurement model (CTO owns technical metrics; CFO owns financial outcomes), cited in Quartz, June 11, 2026.

[21]BCG research on "AI future-built" companies: five times the revenue gains and three times the cost reductions of peers, driven by clear outcomes rather than higher spend. Cited in Quartz, June 11, 2026.

The Window

The danger is not overspending. It is overcorrecting. A cap that hardens into a permanent budget locks in last year's ceiling and quietly forfeits the upside, right as the tools get better and cheaper. The companies that win this phase will not be the ones that spent the least. They will be the ones that stopped counting tokens and started counting what those tokens produced.

That is a finance problem before it is an engineering one, and it is a familiar one: a new, fast-growing, poorly-instrumented line item that has to be measured against the work it replaces. Finance teams have done this before. AI is just the newest version, and it happens to be the one most likely to pay for itself when you measure it honestly.

Rillet is the only ERP built on real-time architecture, used by 600+ finance teams. We track our own AI spend the same way we tell customers to: against the work it replaces.

Rillet

Rillet is the AI-native ERP built by accountants for accountants, designed from the ground up around Continuous Close architecture. To learn more, visit rillet.com.