

QuantumBlack, AI by McKinsey and McKinsey Technology and Tech & AI Practices

Cost versus value: Managing agentic AI system performance

Per-token pricing has stopped being a useful measure for what enterprises actually pay for gen AI. A conversation with David Tepper, CEO of Pay-i, delves into the new economics of measurement.



For the first two years of gen AI adoption, most enterprises focused on access, experimentation, and deployment. As agents move into production, a different set of questions is emerging: How should leaders think about the economics of systems whose costs scale with usage rather than users? What happens when information, software, and even coding itself become increasingly generated on demand? The decision to scale an agent is increasingly becoming a complex and fast-changing economics decision, not a technical one.

To explore these ideas in greater depth, McKinsey Senior Partner [Lari Hämäläinen](#) spoke with David Tepper, CEO and cofounder of Pay-i, who has spent the past three years building measurement and economics infrastructure for enterprise agentic systems (see sidebar “About Pay-i”). Tepper spent 19 years at Microsoft, where he held leadership roles related to Azure’s internal gen AI consumption strategy, and holds an early gen AI patent dating back to 2011.

About Pay-i

Pay-i is a Seattle-based software company focused on measuring and optimizing the economics of gen AI systems. Founded by former Microsoft executives David Tepper, Doron Holan, and Erik Winters, the company provides tools that help enterprises track AI usage, connect model activity to business KPIs, and manage AI-related costs across applications and agents. The company raised \$8.2 million in seed funding in 2025 and positions its platform at the intersection of AI observability, FinOps, and ROI measurement for enterprise AI deployments.

Over a series of conversations, they discussed the changing pricing dynamics of AI, the emerging economics of agentic systems, and where the future of AI economics is headed.

The conversations have been edited for length and clarity.

Learning that agentic economics are different from cloud

Lari Hämäläinen: David, could you briefly describe the journey that led you to found Pay-i?

David Tepper: I came at this problem from the inside. I ran internal gen AI consumption for Azure right as ChatGPT launched, which meant managing GPU utilization and capacity across divisions that were each spending \$250,000 a day on AI and tracking it all in Excel because

there was no other way to do so. One day, I went around asking questions, made some tweaks, and went back to my boss and said: I just saved \$300,000 a week in perpetuity. That was the moment I realized the financial realities of gen AI are grossly different from those of cloud. Cloud has a high up-front cost, and then each new user adds costs slowly; the entire SaaS economy is built on that scaling. Gen AI is a consumption economy from the first call.

The second thing I learned came from building Pay-i. We went two years with people telling us we were crazy. Token prices were falling, and everyone said the bill was going to zero. Then the inflection arrived: Even as prices fell, what enterprises asked the technology to do grew by orders of magnitude more. Tokens are not value. Tokens are the bill. The bill tells you what you spent. It does not tell you whether you should have spent it. That gap, between the bill and the business outcome, is most of what we set out to close, and it is the world I want to talk about.

‘Which of my 47 agents are worth it, and how would I know?’

Lari Hämäläinen: When a CIO or CFO calls you today, what is the conversation they are actually having internally that brings them to your door?

David Tepper: Eighteen months ago, the call came from an engineering leader who wanted to instrument cost. Today, it is from a CFO or a CIO, and they are not asking how to reduce AI spend. They are asking how to justify it. Leaders are getting hit with the reality of managing the fastest-growing new cost center in the enterprise—maybe the fastest growing in history.

The reality is that most enterprises have not built the measurement architecture that would let them tell a real story to a board. They have a Copilot rollout, a customer service bot, multiple coding assistants, and probably six or seven other things their business units stood up without telling central IT. I call that “agent sprawl.” You can see it in engineers posting \$100,000 token bills on LinkedIn almost as a badge of honor. We’ve token-maxed like crazy, but there is no line that says what the business got for it.

There is real value being generated in a lot of these deployments; the enterprises just have no instrumentation to see it. And it is not only a finance problem. The engineering leader who cannot prove an agent is working also cannot defend a road map or kill a failing prototype.

So the question enterprises are starting to ask is no longer “Is AI worth it?” in the abstract. It is “Which of my 47 agents are worth it, and how would I know?” [See sidebar “Six drivers of agentic AI operating expenditures.”]

Why most enterprises measure AI incorrectly

Lari Hämäläinen: When it comes to agentic systems, what are the “laws of nature” that enterprises still do not fully understand?

‘That was the moment I realized the financial realities of gen AI are grossly different from those of cloud.’

David Tepper: The challenge with ROI in agentic systems is that most organizations are still asking the wrong questions. The conversation tends to focus on models and technical optimization, but almost nobody starts with the business KPIs for the use case itself. What are you actually trying to accomplish? What defines success? We speak with hundreds of enterprises, each pursuing dozens of use cases, and when we ask how they define success for an agentic system, people often struggle to answer.

Unlike traditional software, these systems are stochastic. They are not deterministic systems where you input two plus two and reliably receive four. It is much closer to managing a human employee. If you cannot define those outcomes, organizations end up prematurely optimizing technical details without understanding whether the system is actually delivering business value.

Traditional software KPIs focus on performance metrics such as run time and failure rates. But when systems are dynamic and making decisions at run time, the KPIs become business oriented: What outputs are being generated, what business outcomes are improving, and how do you measure that impact? That structured way of thinking about success is still missing in most organizations.

Organizations seeking to maximize AI’s impact need observability and governance at that core layer, which is distinctly different from traditional developer-oriented observability. They need to understand what employees and agents are doing, what business outcomes they are trying to achieve, and whether the value generated exceeds the cost. Once you have that framework, you can A/B test different approaches, compare human-only teams with agent-assisted teams, and identify what actually works. The biggest pitfall is failing to quantify the experiments.

Workflow, pipeline, agent: Three different problems hiding under one word

Lari Hämäläinen: Before we get to evaluation, let’s address a definitional question. Everything is being called an agent right now. How do you cut through that for an executive?

David Tepper: It matters more than people think, because the cost problem and the evaluation problem look genuinely different across three categories that all get called agents.

Six drivers of agentic AI operating expenditures

In McKinsey's experience, six issues drive most of the cost with agentic AI:

- *Long-lived context.* Agentic tasks can consume about 1,000 times more tokens than code-reasoning (single-turn problem-solving without tool interaction) or chat tasks (multi-turn dialogue about a coding problem).¹ Context becomes an operating asset and accounts for a significant portion of agentic costs.
- *Refinement is the sink.* In agentic workflows, the expensive part is not the first answer generated but the checking, repairing, and re-verifying that follows. About 60 percent of an agentic task's costs, in fact, are tied to refining answers.²
- *Autonomy creates variance.* Agentic systems can produce materially different costs for the same task because they may take different paths, call different tools, or retry in different ways. Cost behaves as a distribution, not a fixed unit price.
- *Expensive reasoning is used for basic tasks.* Extended thinking pays for itself many times over on hard tasks. For easy ones, it is expensive overhead.
- *Agent choice orchestration can compound costs.* How an agent calls a tool matters as much as what it calls. The way work is decomposed, coordinated, and handed off across agents, tools, and models can change costs by an order of magnitude without changing the business outcome.
- *Information structure can drive inefficiencies.* How efficiently information is presented to or by a model (for example, prompt design, context length, language, formatting, and data structure) affects token consumption. Non-English text, for example, gets fragmented into more tokens per meaning, so the same conversation costs more in some languages than others.

¹Bai et al., "How do AI agents spend your money? Analyzing and predicting token consumption in agentic coding tasks," arXiv, April 2026, arXiv:2604.22750; "How we built our multi-agent research system," Anthropic, June 13, 2025.

²Mohamad Salim, Jasmine Latendresse, Sayed Hassan Khatounabadi, and Emad Shihab, "Tokenomics: Quantifying where tokens are used in agentic software engineering," arXiv, January 2026, arXiv:2601.14470. Analysis of 20 production agentic coding workflows on SWE-bench and internal benchmarks. The 59.4 percent review/refinement share and 53.9 percent average input-token share are direct findings. Coding is the first scaled laboratory for agentic token economics; the mechanism (verification dominates creation) generalizes across enterprise workloads, but exact ratios will vary with workload type and verification depth.

A workflow is traditional software with gen AI bolted into one or more steps. A pipeline is a series of predetermined steps that call an LLM [large language model] at one or more of them. Most chatbots are pipelines. A truly agentic system is given a broad objective, has tools and inputs available, and decides at run time what to call, in what order, and for how long. Coding assistants are the most prevalent of these today.

The distinction matters for executives for a number of reasons. For one, cost behaves differently: A workflow is bounded per invocation, a pipeline is bounded per call times a known number of calls, and a true agent is unbounded per task, which is why there can be a 30 times variance on outcomes from the same prompt. Another reason is that evaluation breaks at different points: Workflows can be tested with a simple test set; pipelines mostly survive conventional “LLM-as-judge” methods because the trajectory is fixed. However, true agents cannot be evaluated in this way. Most vendors selling you an “agent” are selling you a pipeline. That is often exactly what you need, but you should know what you are buying.

Why LLM-as-judge stops working the moment agents get interesting

Lari Hämäläinen: OK, with that taxonomy in mind, where does the evaluation technique of LLM-as-judge stop working?

David Tepper: It works fine for simple workflows and pipelines, where you have a relatively canned prompt and reference answers to score against. It breaks for true agents, and it breaks in an uncomfortable way. Agents introduce randomness across many turns, not just one. And agents construct their own prompts internally as they go, so you cannot build a database of acceptable inputs and outputs, because the inputs change every run.

The factor-of-30 variance in token consumption is what tells you the rest of the story. The trajectory is wide while the end point is narrow. Most of the signal lives in the trajectory, and a judge that only looks at the final output is grading the wrong artifact. When an agent fails, it almost never fails on the last step. It fails three steps earlier, potentially in a way the last step covers up. If you only judge the output, you are grading the cover-up, not the crime. Companies today are typically only using traces at the engineering level to optimize the code. Most have not yet implemented systems that identify the business impact of the decisions the agents are making.

The fix for evaluating agents is to anchor to business outcomes, not just to the raw model outputs. For cases where the agent achieved a defined business outcome, you extract the core behaviors from traces of the entire use case, not just a single request or session. You collect a corpus of these behaviors in the same way you would store a database of known good prompts and responses. You do the same thing for cases where the agent did not achieve the business outcome, essentially giving you the ability to do counterfactual credit assignment.

In practice, the corpus is not a static asset. The same agent against the same input will produce different traces tomorrow, so it needs continuous curation; a versioning scheme tying each trace to the prompt, model, and tool definitions in effect at the time; and an answer to what carries over when a new frontier model arrives.

‘When an agent fails, it almost never fails on the last step. It fails three steps earlier, potentially in a way the last step covers up.’

The math that decides whether to deploy an agent at all

Lari Härmäläinen: You have argued that the case for deploying an agent reduces to a fairly clean math question. Can you walk us through it, and where is it more subtle than it sounds?

David Tepper: The formulation is straightforward. An agent adds value when the probability of success is greater than the time it takes a human to verify the work divided by the time it takes a human to do the work themselves. $P(\text{success})$ is greater than $T(\text{verify})$ over $T(\text{do})$.

Run the math on a real case. Imagine a task takes a human two hours to complete but only six minutes to verify whether the agent did it correctly. $T(\text{verify})$ divided by $T(\text{do})$ is about 5 percent. That means the agent only needs to succeed five times out of a hundred to be net positive. For a large class of enterprise work, the inequality is far more forgiving than people assume.

The trap is that this clean version of the math only holds when failure leaves the environment unchanged. A document-processing agent that gets it wrong produces a bad document; you discard it, no harm done. A customer service agent that tells a customer they can refund a nonrefundable trip has changed the environment. Now you have a recovery cost on top of the original cost, and the cost of failure exceeds the cost of doing the work. Some call this the agency tax: verification cost plus rework cost, and in production, rework is rarely zero. The required reliability bar jumps sharply, and most enterprises have not done that calculation use case by use case.

This is also why frontier model adoption is rational at a higher per-token cost. A more capable model lifts $P(\text{success})$ and often shrinks $T(\text{verify})$ at the same time. If a more expensive model takes verification from ten minutes to two, the productivity gain almost always swamps the additional token spend. The deeper implication is that the cost of a gen AI use case is no longer a number. It is a distribution with likelihoods, expected values, and percentiles.

One last point. ROI is not a KPI. ROI is what you compute from KPIs and from cost. We have talked to hundreds of enterprises, each with dozens of use cases, and when we ask how they

define success for an agentic system, they stare at us blankly. Without that framework, everything you do at the inference layer is premature optimization, which any engineer will tell you is the root of all evil.

The end of the per-token price decline

Lari Hämäläinen: You have written that the comforting story about AI getting cheaper is true in a narrow sense and misleading in a broader one. What is actually happening?

David Tepper: Pay-i has tracked inference cost for a fixed model size compounding down roughly 6.67 percent per month since 2022, which works out to a decline of about 86 percent for models of similar size. If you hold capability constant, the price falls. That is the story most CFOs have heard, and it is the story most providers are happy to tell.

The challenge is that nobody is holding capability constant. Agents are the multiplier. A single agentic task can comprise dozens or hundreds of API calls, and the industry average inside an agent today is about three-and-a-half different models per task, frequently from different providers. That is not three and a half across an enterprise. That is inside one agent's run. The Microsoft and Stanford finding that agentic workflows consume roughly 1,000 times more tokens than comparable code chat and code-reasoning sessions is consistent with what we see in production telemetry.

Context windows have also grown roughly 250-fold since GPT-3, so the cost of a call depends more on how full the window is than on the headline rate, and the bulk of the cost is on the input side. Most of the bill comes from what you have asked the model to read, not from what it says back. The metric that matters is cost per completed task, and that number has gone up steadily in every workload we see because the appetite to leverage the technology for increasingly involved tasks is insatiable.

The hidden cost layers compound this. Cached input tokens are 75 to 90 percent cheaper, but only if the prompt is structured to hit cache, and most are not. Reasoning tokens can dwarf visible output.

The capacity cliff, and the latencies most teams forget to measure

Lari Hämäläinen: At a certain scale, the conversation stops being about token pricing and becomes about capacity. When does that switch happen?

David Tepper: Roughly \$3 million of annual spend concentrated with a single provider for a single model. Below that, shared infrastructure works fine. Above it, time-outs, rate limits, and shortages start hitting production, and you are forced to move to provisioned capacity. That threshold is not a hard rule, but it is where most of our customers cross over.

‘An agent adds value when the probability of success is greater than the time it takes a human to verify the work divided by the time it takes a human to do the work themselves.’

The thing nobody warns you about is that provisioned capacity does not degrade gracefully; everything runs cleanly right up until it does not, and the failure usually arrives when a new agent or team is being onboarded. So the move to provisioned capacity is a graduation. It is the moment your gen AI application becomes mission critical, and the moment your variable-cost line becomes a fixed-cost line, with all the planning discipline that implies.

The transition rewards preparation: load-test against your actual production traffic mix well before you cross the threshold, and negotiate what happens when a newer model becomes available mid-commitment.

The most common failure mode is siloing. A defensive engineering team provisions a separate pool per use case so no agent can starve another. This destroys the economic logic of provisioning, which depends on diverse workloads sharing peaks. We see enterprises paying for two or three times the capacity they need because of this one mistake.

One last thing. The TPM number, tokens per minute, that providers advertise on these reservations is a rough estimate, not a load specification. Real capacity depends on the input–output mix, the caching strategy, and request-per-minute ceilings that providers usually do not publish. The only honest way to size a reservation is to load-test your actual workload.

The playbook for the next four quarters

Lari Härmäläinen: If you were sitting across from a CIO and a CFO, and they wanted a short list of things to do in the next two to four quarters, what would you tell them?

David Tepper: Seven moves, in roughly this order.

First, inventory the agents and split them into use cases. Most enterprises cannot produce an accurate list of agents in production, the business owner for each, and the KPI each is supposed

to move.

Second, define quantifiable KPIs and intended business outcomes up front. How will you measure success in business outcomes, and what do those outcomes translate into in hours saved, dollars generated, or error rate reduced? Get the business sponsor to sign their name to it before launch. Retrofitting KPIs after launch is where attribution dies.

Third, instrument to measure all business levers, not just the bill. Token spend by team is table stakes and insufficient to catch outliers or to measure business value. The most valuable instrumentation is at the workflow and agent-step level, where you can attribute not only cost and latency but also outcomes. Without it, you can't determine whether an agent is improving productivity, and the causal claims a CFO needs cannot be made.

Fourth, decide who owns the measurement architecture. The most common organizational mistake I see is leaving this ambiguous. Pick one to own it explicitly and resource the gap: engineering head count for FinOps, a finance partner for the COE [center of excellence], a business sponsor for the platform team.

Fifth, treat capacity as a portfolio. Do not silo provisioned pools per use case unless there is a regulatory reason. Mix workloads with different traffic shapes across shared pools, and use spillover deliberately rather than as a failover accident. The same portfolio logic applies to model choice: The right answer is workload by workload, not one answer enterprise-wide.

Sixth, run a quarterly model review. Frontier models move every three to six months. Decisions made in quarter one will be wrong by quarter three. I can't tell you how many enterprises we talk with are still using GPT-4.1 when GPT-5.5 is available.

Seventh, stand up a joint operating group across finance and engineering and a business sponsor for every material agent. Without it, the CFO defends spend without context, the engineer defends models without business framing, and the business owner cannot measure business value. Gen AI is a different business model, and it requires teams to work together in new ways to optimize its impact.

What the conversation will be about in 12 months

Lari Hämäläinen: Look out a year. What is the conversation a CIO and CFO will be having that they are not having today?

David Tepper: I think bifurcation will start to show up in customer-facing metrics. As Mythos-class models reach production, the gap between enterprises that can deploy them and those that cannot will compound. The board question shifts from "Are we using AI?" to "Are we using the right AI?" and the enterprises that bet on the wrong model tier in 2026 will feel it.

Find more content like this on the
McKinsey Insights App



Scan • Download • Personalize



Then the observability and FinOps layer will become standard procurement. The analogy I use is APM [application performance monitoring] in the 2010s: The category did not exist three years ago and will be a line item in the typical enterprise stack in 18 months.

Similarly, the changes already happening with agentic coding are a harbinger of what's to come for many other white-collar professions. If there is intelligence in the tokens, so to speak, then maximizing the use of that intelligence by token-maxing is what will naturally happen as tools and awareness spread to other employees.

Beyond all of that, the AI bubble narrative will quietly die. There is no bubble bursting here. There is a category in which 95 percent of enterprises are not measuring their returns, in which inference revenue is compounding at a rate no prior software category has matched, and in which capability improves on a three- to six-month cadence. The enterprises that mistake “We cannot see the ROI” for “There is no ROI” will look back on this period the way the laggards looked back on cloud in 2014.

David Tepper is CEO and cofounder of Pay-i. **Lari Hämäläinen** is a senior partner in McKinsey's Seattle office.

Comments and opinions expressed by interviewees are their own and do not represent or reflect the opinions, policies, or positions of McKinsey & Company or have its endorsement.

This interview was edited by Barr Seitz, an editorial director in the New York office.

Copyright © 2026 McKinsey & Company. All rights reserved.