

GRID DYNAMICS EBOOK

Why AI must target constraints in the value chain to deliver real ROI

Contents

Introduction 3

Where should AI be applied to move the numbers that matter? 4

An enterprise decision framework for agentic AI transformation 5

What constraint impact should you invest in? 8

Conclusion 13

About Grid Dynamics 15

Introduction

Corporate spending on agentic AI has surged into the hundreds of billions of dollars. Yet the effect on core business performance remains surprisingly thin. In 2026, only 30% of enterprises reported measurable financial impact from their AI programs. Many organizations can point to pilots, prototypes, and pockets of time savings, but far fewer can show that claims settle faster, orders move through plants more quickly, or service costs fall without harming quality. The productivity boom widely associated with AI has barely appeared in the profit and loss (P&L) statement.

The mismatch is not mainly about technology; it is about failing to embed AI into core value-creating processes. Most initiatives target individual tasks — drafting emails, summarizing documents, generating code — rather than the end-to-end flows that determine whether value is actually created. In systems where a single slow step governs the pace of everything else, accelerating surrounding work rarely improves outcomes. Instead, it often produces longer queues and additional handoffs.

This is the central challenge of agentic AI business transformation. Organizations may save roughly 25 minutes per employee per day using AI, yet only 3-7% of that productivity typically translates into earnings or EBIT improvement, as the remainder is absorbed by rework, oversight, and process friction.

This pattern is visible across sectors. In procurement, AI auto-classifies invoices and expenses faster, yet payment approvals still wait in overloaded manager queues, leaving days payable outstanding and supplier satisfaction unchanged. In manufacturing, AI forecasting sharpens demand signals, but production changeovers and maintenance scheduling remain constrained, so stockouts and rush orders continue.



Where should AI be applied to move the numbers that matter?

The more pragmatic starting point is to identify where work actually gets stuck — and why. In most organizations, performance is limited by a specific constraint in the value stream — a step, role, policy, or capability that throttles the entire system, such as a senior reviewer, an overloaded decision gate, or a downstream control that routinely holds up release. When attention and investment concentrate on relieving that choke point — and on the few steps that feed into and out of it — throughput rises, queues shrink, and additional capacity becomes real.

This dynamic is explained by the Theory of Constraints (TOC). In any value stream, overall performance is governed by the current constraint — the point that limits throughput. Improvements away from it rarely translate into better outcomes; instead, they appear as waiting time, coordination overhead, or rework. TOC prioritizes interventions that increase flow through the constraint. Organizations first exploit the constraint by removing avoidable losses such as administrative work, setup time, rework, and decision delays. Next, they subordinate surrounding steps so the constraint is consistently fed with ready-to-process work and is not blocked from releasing output. Only then does it make sense to elevate capacity, recognizing that the constraint may simply shift elsewhere.

Seen this way, agentic AI for organizational transformation is less like a magic wand and more like an amplifier for basic operational discipline. When applied to remove avoidable losses at the bottleneck, stabilize work around it, and clear blocks that prevent completed work from turning into cash, it can move the measures that matter: units shipped, cases closed, revenue collected, and costs reduced without breaking service.

To help leaders evaluate where AI should be applied for measurable results, Grid Dynamics proposes an agentic AI enterprise transformation decision framework focused on return on impact (ROI) and alignment with end-to-end business outcomes.

An enterprise decision framework for agentic AI transformation

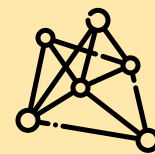
The practical rule is simple: start narrow, prove impact, then scale. Instead of chasing “AI everywhere,” focus on three to five processes where better flow would materially change business outcomes. Expansion should follow demonstrated results, not assumed capability.

To determine where AI-enabled improvement should be deployed, we follow a simple model:



Discover →

Discover how work actually flows today. Event logs, workflow tools, or even a small set of case samples can reveal where time accumulates. Together, these sources highlight the real constraint and the handful of steps that feed it, block it, or send work back to it.



Model →

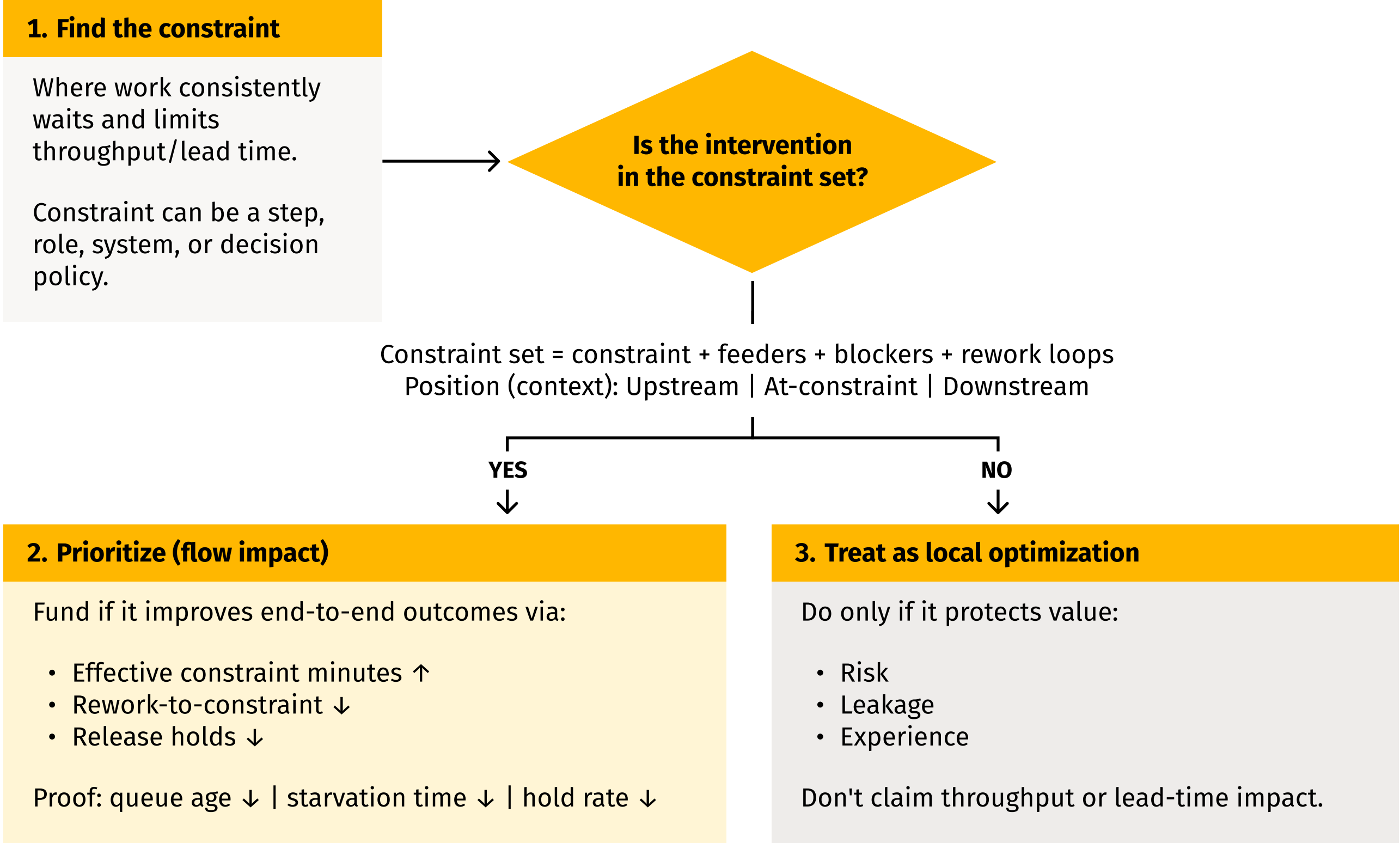
Model lightweight simulations or “what-if” analyses to test how different interventions would affect throughput and lead times, and whether speeding up one step would simply shift congestion elsewhere in the process.



Invest

Invest in initiatives that show a clear mechanism for improving system outcomes: more productive minutes at the bottleneck, less rework returning to it, fewer completed items stuck downstream, or cost reductions that still hold as volumes rise. The metrics used to prove that impact should be defined from the start.

Agentic AI enterprise transformation decision framework



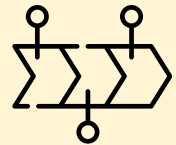
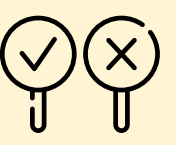
Feeder = prevents starvation | Blocker = prevents release | Rework loop = returns work through the constraint

Agentic AI investment decision logic

The framework appears straightforward. Yet the shine of AI and the promise or fear of disruption often distracts organizations from disciplined decision-making.

The Discover → Model → Invest sequence only works when improvements demonstrate how they change system flow, rather than simply increasing local activity.

To evaluate this properly, classify the step and its AI intervention relative to the constraint in the workflow — upstream, at the constraint, or downstream — and then determine how the intervention affects the constraint itself.

 Where the step sits in the flow	 Default decision bias	 How do you prove value (business-first, metrics optional)
Before the bottleneck step (work has not yet reached the step that sets the overall pace)	Usually NO (risk of local optimization) YES only if it: keeps the bottleneck supplied with “ready” work or prevents errors that return later	• fewer “waiting for ready work” moments (bottleneck downtime ↓) • fewer errors returning to the bottleneck (returns/rework ↓)
At the bottleneck step (the step that sets the overall pace)	Usually NO (risk of local optimization) YES only if it: keeps the bottleneck supplied with “ready” work or prevents errors that return later	• throughput ↑ (units /cases/orders per week) • less admin, setup, or decision delay at the bottleneck (productive time ↑)
After the bottleneck step (work is “done” but not yet shipped, billed, or closed)	Usually NO (unless value is getting stuck) YES if it: unlocks shipment, billing, or closure, or prevents revenue leakage	• “ready but stuck” time ↓ • fewer write-offs, penalties, disputes, or compliance misses

What constraint impact should you invest in?

Evaluate agentic AI opportunities by how they affect the constraint in these four ways.

Increase effective constraint minutes: Time saved at the bottleneck or time recovered from avoidable losses around it translates most directly into additional throughput.

Reduce rework returning to the constraint: If work repeatedly returns to the bottleneck because of missing information, exceptions, or quality defects, improving first-time-right flow increases productive constraint time and shortens lead time.

Reduce downstream blocking of release: Sometimes output exists but cannot be released due to approvals, compliance checks, billing disputes, or downstream capacity gaps. Removing these blockers converts production into realized value.

Deliver cost removal that survives a throughput stress test: Treat “time saved” as capacity until you can prove that costs can be removed —headcount, overtime, contractors, or paid capacity — without degrading service. A throughput stress test serves as a practical safeguard. Assume the initiative succeeds and throughput increases materially, say by 20–30%. If queues form or utilization spikes elsewhere, the constraint has simply moved. In that case, the “savings” are not yet bankable.

Everything else should be treated as a local productivity improvement. It may still be worthwhile for experience, compliance, or leakage prevention, but it should not be expected to improve throughput, shorten lead time, or deliver durable P&L impact unless it connects directly to one of the value paths above.

For example, in a discrete manufacturing plant, the true constraint may be a chronic machine changeover step rather than overall line capacity. An agentic AI initiative that standardizes kitting, generates optimized changeover sequences, and orchestrates maintenance windows increases effective constraint minutes, reduces rework returning from startup failures, and prevents finished orders from sitting “ready but not shipped” while quality or documentation catches up. Once throughput rises and a stress test shows the system can sustain 20–30% higher volume without new bottlenecks, cost removal (overtime, contractor spend, or redundant capacity) becomes credibly bankable.

WHY: Discover the constraint		WHAT: Model the solution	HOW: Invest in design & execution
Before • Periodic reports flag issues late • Multi-week consultant diagnostics • Data siloed across functions Example • Plant productivity decline appears in quarterly operations review • 8-week consulting engagement interviews 40+ stakeholders to find the bottleneck • Event logs and changeover data scattered across ERP, MES, and maintenance systems	After • AI reconstructs how work actually flows today from event logs, workflow tools, and case samples. • AI highlights where time really accumulates and which steps feed, block, or send work back to the constraint. • Human leaders reframe the strategic question around the true constraint, not assumed line capacity. Example • AI analyzes production event logs and identifies machine changeover, not material flow, as the true constraint • AI surfaces patterns showing that changeover delays cause 40% of planned downtime • Human plant leadership reframes AI roadmap around reducing changeover time instead of adding line capacity	Before • Small group brainstorm behind closed doors • Options filtered by experience bias • Decision criteria implicit, undocumented	Before • Large teams, months-long planning • Manual decks & static spreadsheets • Serial committee reviews slow pilots
		Example • Operations VP proposes "lean manufacturing" workshops based on prior transformation playbook • Three kaizen events map current state across prep, changeover, and startup phases • Shortlist prioritizes operator training and 5S, with little discussion of constraint impact	Example • 80-person transformation program; 12-month "future state" design • Consultants build static process maps and business cases over 6 weeks • Pilot runs in one plant with monthly steering reviews; lessons don't update the operating model
		After • AI generates a wide option set tied to the constraint diagnosis, then runs lightweight "what-if" simulations. • AI estimates how each intervention affects throughput, lead time, and WIP at and around the constraint. • Human teams adjust weights live (throughput, lead time, risk) and see where congestion would simply move elsewhere.	After • AI generates slides, maps, and playbooks that tie each initiative to a clear mechanism: more productive minutes at the bottleneck, less rework, or fewer items stuck downstream. • Human teams lock in a small set of metrics from the start (throughput at constraint, rework returning to it, downstream WIP, unit cost at target volume). • AI tracks pilot results and flags when improvements are robust enough that cost removal (overtime, contractors, redundant capacity) is credibly bankable.
		Example • AI simulates bundles of interventions: standardized kitting, optimized changeover sequences, maintenance windows, and downstream QA/documentation changes. • The system stress-tests whether a 20–30% volume increase keeps the changeover step the constraint or pushes congestion into QA or shipping. • Human leadership prioritizes the option set that increases effective constraint minutes and avoids creating a new bottleneck downstream.	Example • AI drafts the changeover workflow redesign, kitting checklist, and co-pilot prompts, then orchestrates an instrumented pilot. • Within weeks, the plant shows faster changeovers, higher effective constraint minutes, fewer startup failures returning to rework, and fewer finished orders stuck waiting on QA or documentation. • After simulations and pilot data confirm the system can sustain 20–30% higher volume without new bottlenecks, finance and operations treat overtime and excess capacity reductions as bankable savings.



Why upstream productivity can be misleading

Consider how easily organizations misread upstream productivity as progress.

In a claims operation, for example, intake summarization and data capture may be automated so the front office can submit more claims per day. Intake metrics improve immediately. Yet if senior adjuster review is the real constraint, the outcome is predictable: more arrivals hit the adjuster queue without improving file completeness. Queue age grows, exceptions increase, and more claims bounce back for missing evidence.

This is a classic example of upstream optimization outside the constraint, which increases work-in-progress and lead time without improving throughput.

Upstream investments that support the constraint

Now compare that with an upstream investment that sits inside the constraint set.

In the same claims operation, adjusters might occasionally sit idle because files arrive incomplete or essential documentation is missing. The highest-return intervention is not increasing intake volume but improving case readiness — pre-validation, document chasing, and “kitting” the file so that when it reaches the adjuster, it is decision-ready.

The economics change because the lever changes. Instead of increasing arrivals, the organization reduces starvation and rework to the constraint, increasing productive constraint minutes.

Intervening directly at the constraint

At the constraint itself, the logic is the most direct. When utilization is high and queues persist, minutes recovered at the bottleneck translate almost one-for-one into higher throughput until the constraint shifts. Organizations should prioritize interventions at the constraints that:

- remove non-value-added work
- eliminate delays
- reduce rework
- stabilize work content

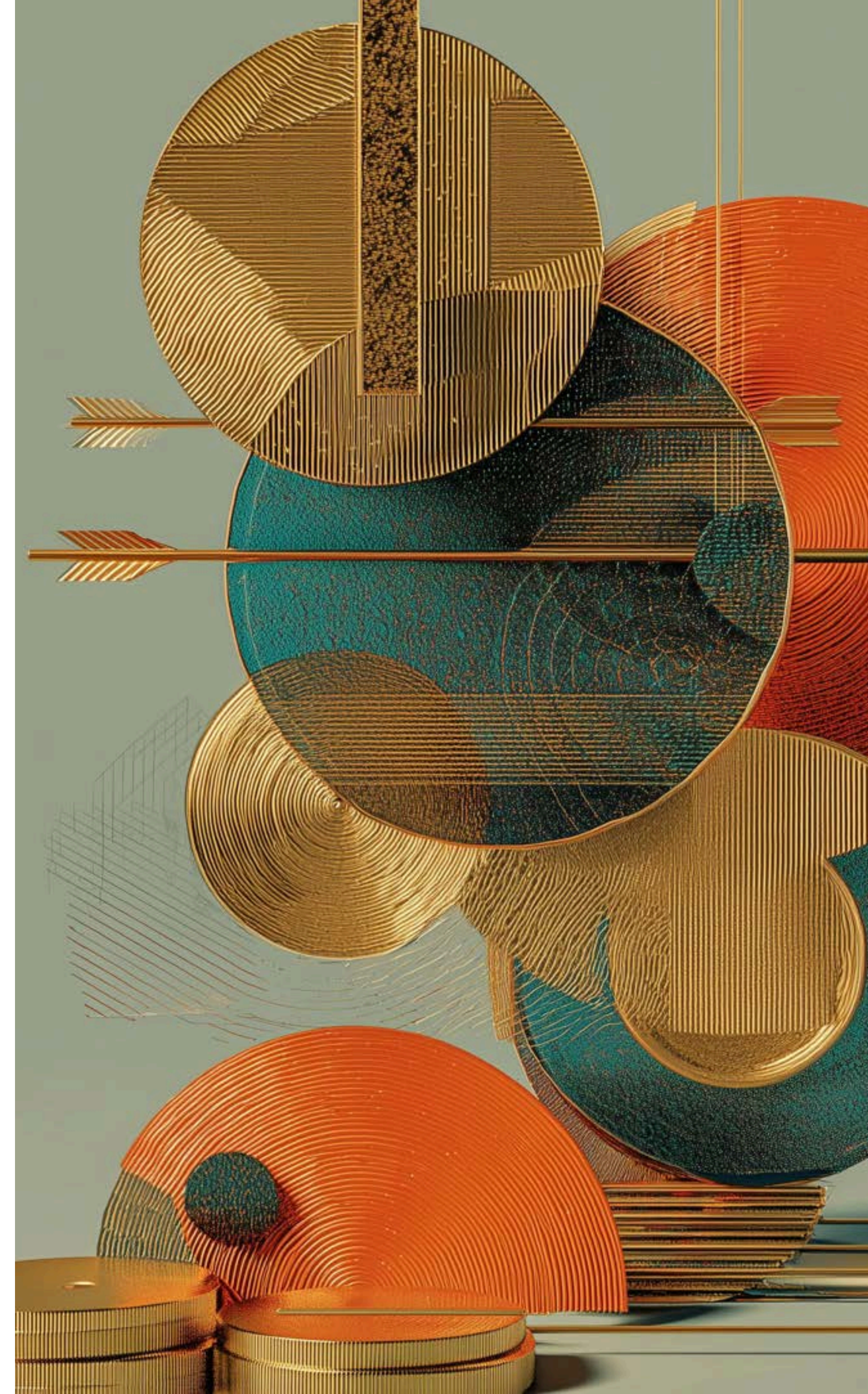
Even improvements that do not involve automation can generate measurable financial impact.

Downstream efficiency vs value realization

Downstream is where organizations most often confuse efficiency with value realization.

If downstream work sits outside the constraint set—formatting, reporting, or post-processing after value is already realized — it may improve experience but rarely changes throughput or lead time.

Downstream investments become attractive when they prevent measurable leakage, such as billing disputes, write-offs, or penalties, or when downstream steps actively block release. In those situations, the value lies in converting completed work into delivered, billable, or compliant outcomes.



The cost-takeout trap

The cost-takeout rule exists because organizations frequently mistake capacity creation for savings. A practical safeguard is a throughput stress test. Assume constraint improvements succeed and throughput rises materially. If utilization spikes or queues form elsewhere, removing cost in non-constraint areas will simply recreate delay somewhere else in the system.

Until the system demonstrates headroom after the constraint moves, treat benefits as redeployable capacity, not bankable savings.

When productivity gains trigger the wrong decision

This is also where finance leaders can be pushed toward the wrong lever. When productivity rises but flow performance does not, the most visible way to “realize” gains is headcount reduction.

Layoffs may create short-term savings, but they often remove the slack needed to absorb higher throughput, clear downstream holds, or stabilize quality. If the constraint simply shifts — to QA, release management, product decisions, or customer adoption — cutting upstream capacity will recreate queues and extend lead times.

The safeguard remains the same: treat time saved as capacity until the constraint moves and the end-to-end system can sustain higher throughput without creating new congestion. Only then is cost removal likely to hold without degrading service.

Conclusion

Most AI programs don't fail because the tools are weak. They fail because leaders mistake local productivity gains for system-wide performance.

Automating a step, generating outputs faster, or reducing effort in one part of the workflow may look like progress. Yet if the constraint in the system doesn't change, the result is predictable: more work-in-progress, longer queues, and little measurable improvement in throughput or business outcomes.

Treat enterprise AI as an operating-model intervention, not a software rollout. That means avoiding three common traps.

1. Don't accelerate work upstream unless you know the constraint and can demonstrate the system can absorb additional flow. Otherwise, the result is simply more waiting and rework.
2. Don't treat "time saved" as savings. Until costs are removed or capacity converts into higher throughput, time saved is only potential capacity.
3. Don't scale pilots by copying point solutions across the organization. Real transformation requires redesigning roles, controls, handoffs, and governance so the entire system behaves differently.

Organizations that consistently capture ROI follow a disciplined path: diagnose the constraint, model the flow impact, invest where outcomes move, and stress-test the system before declaring success. This discipline is at the heart of successful agentic AI enterprise transformation — AI becomes measurable performance, not just more activity.



Ready to turn agentic AI enterprise transformation into measurable operational impact?

Start by identifying the constraints in your core workflows and evaluating where intelligent agents can remove friction, reduce rework, or unlock throughput. Grid Dynamics can help you assess where AI agents will create the greatest business impact, and how to scale them safely across the organization.

SIGN UP FOR A FREE CONSULTATION →

About Grid Dynamics

Grid Dynamics empowers enterprises to drive growth, boost efficiency, and transform their digital capabilities through expert technology consulting, platform engineering, AI, and advanced analytics.

Fusing technical vision with business acumen, Grid Dynamics (NASDAQ: GDYN) fuels a relentless drive toward sustainable digital transformation. As a forefront provider of technology consulting, platform and product engineering services, and bespoke software development, we draw from over 8 years of leadership in Enterprise AI, coupled with profound expertise in cloud, data, and advanced analytics.

Our commitment to engineering excellence, R&D leadership, a co-innovation ethos, globally efficient follow-the-sun delivery model, and an unwavering "whatever it takes" dedication to client success empower us to solve even the most complex enterprise challenges, ensuring profitable business outcomes and future-proof growth.

Founded in 2006, Grid Dynamics is headquartered in Silicon Valley and has a global talent pool of intellectually curious problem solvers in offices across the Americas, Europe, and India.

LET'S TALK AGENTIC ROI →





Grid Dynamics Holdings, Inc.

6101 Bollinger Canyon Road, Suite 465,
San Ramon, CA 94583
www.griddynamics.com