TECH mahindra

THE ENTERPRISE AI PLAYBOOK

Governing AI Spend in the Age of Token-Metered Development

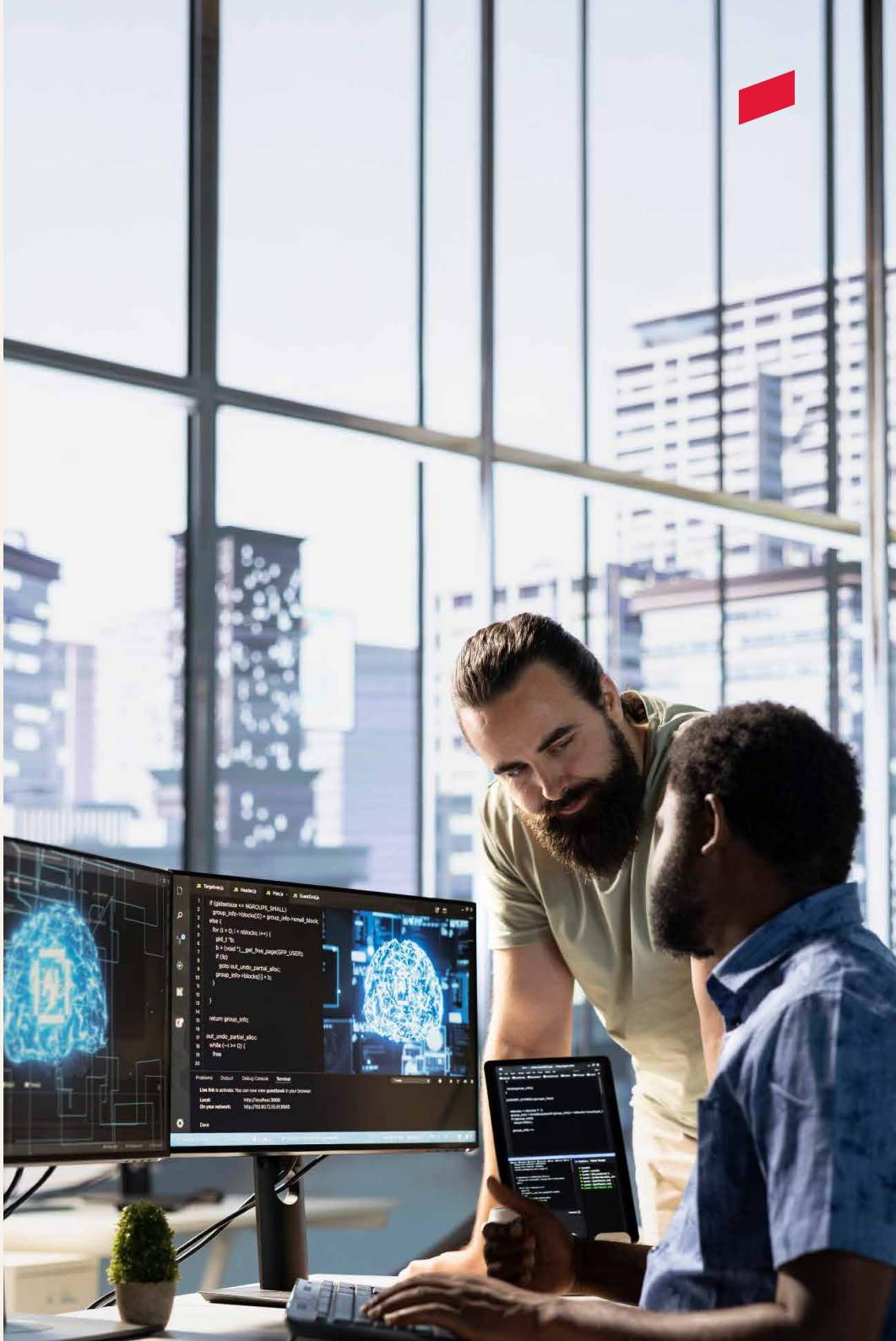
Building an enterprise AI strategy that balances innovation, cost, governance, and developer productivity.

Table of Contents

Executive Summary

Enterprise AI has entered a new economic reality. The shift from predictable subscription pricing to token-metered consumption means AI is no longer a fixed software expense but a dynamic operational cost that scales with every interaction, workflow, and autonomous agent. Engineering leaders must now govern AI consumption with the same discipline applied to cloud infrastructure, cybersecurity, and software delivery.

The playbook examines how organizations can build a repeatable operating model for AI-enabled software engineering. It explores the economics of token-based development, the architectural decisions that drive AI spend, governance mechanisms that improve financial control, and the operating practices that enable engineering teams to scale AI responsibly. It also presents practical guidance for redesigning software delivery, establishing AI finOps, strengthening governance, and measuring business outcomes, helping enterprises transform AI from an experimental capability into an operational advantage.



Introduction

Why This Moment Is Different

On June 1, 2026, enterprise AI development entered a fundamentally different phase.

GitHub Copilot's transition from flat-rate subscriptions to token-metered billing did more than introduce a new pricing model—it changed how organizations must think about software engineering economics.

Most interactive and agentic Copilot capabilities now consume AI credits based on token usage. Code completions and next-edit suggestions remain included and do not consume AI credits on paid plans.
A new update from GitHub specifically states that code completions and next-edit suggestions remain unlimited for paid plans and do not consume AI credits.
AI has shifted from a predictable productivity tool to a variable enterprise resource that requires continuous financial oversight.

This AI transition mirrors the early evolution of cloud computing. Infrastructure was once treated as a fixed procurement expense until consumption-based pricing exposed the true economics of compute. Organizations that successfully navigated that shift developed new disciplines around governance, observability, cost optimization, and FinOps. Enterprise AI is now entering the same stage of maturity.

The challenge, however, extends beyond controlling technology costs. Autonomous coding agents, reasoning models, and multi-step AI workflows are fundamentally changing how software is designed, built, reviewed, and maintained. Traditional engineering practices, delivery metrics, and governance models were not designed for environments where AI participates directly in software development.

> **Gartner predicts that more than 40% of agentic AI projects will be canceled by the end of 2027 because of escalating costs, unclear business value, and inadequate risk controls.**
> **The organizations that succeed will be those that establish governance before AI consumption scales.[1]**

For engineering leaders, platform teams, enterprise architects, and FinOps organizations, this creates an immediate leadership challenge. Success will no longer be measured by how broadly AI is adopted, but by how effectively it is governed, integrated into engineering workflows, and aligned with measurable business outcomes.

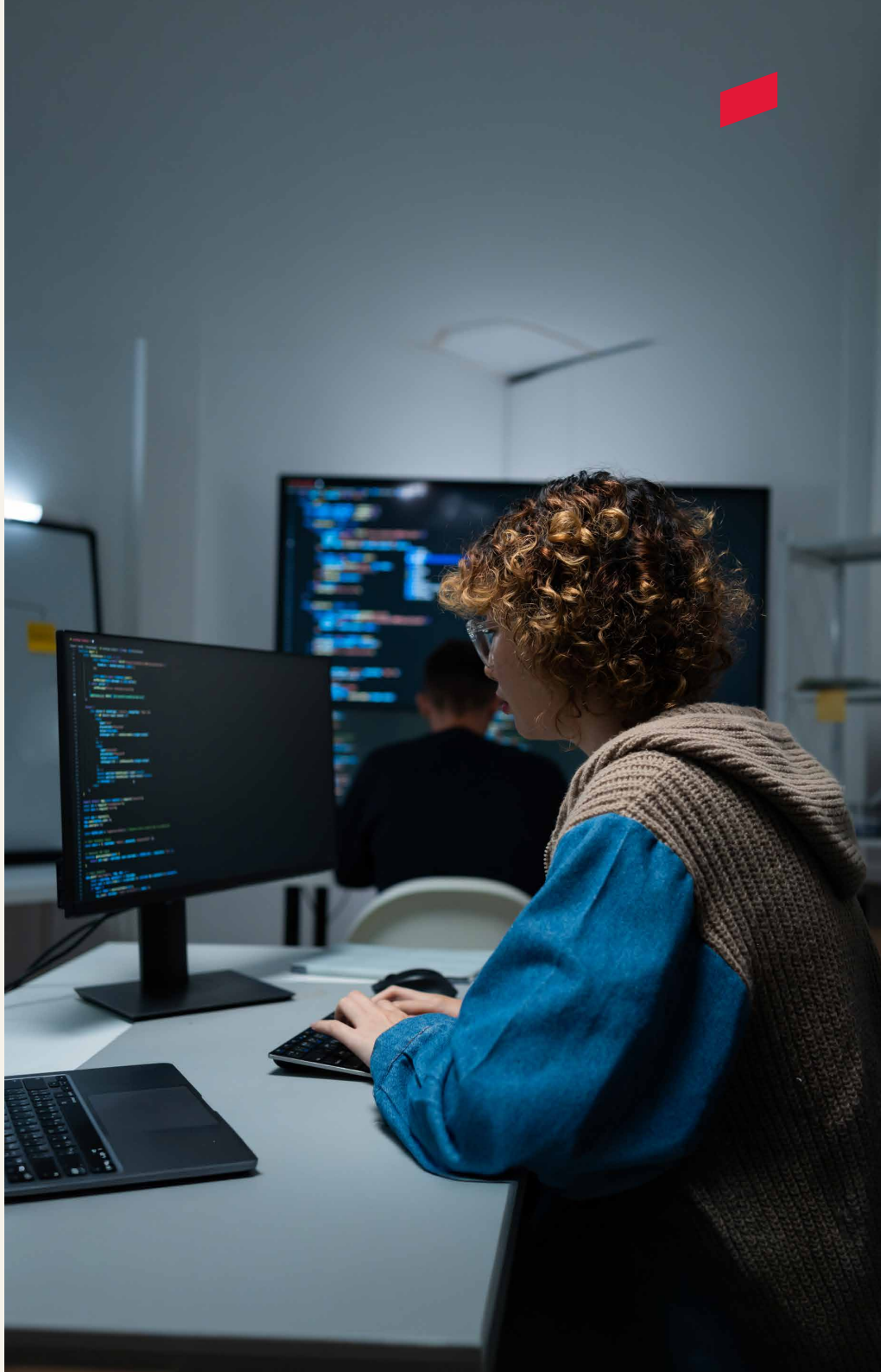
The Billing Model: What Changed and Why It Matters

AI development costs are no longer hidden within software subscriptions. They are visible, measurable, and directly influenced by engineering decisions made every day. Understanding where these costs originate is the first step toward governing them effectively.

From a Flat Fee to AI Credits

Before June 1, 2026, GitHub Copilot Enterprise largely abstracted AI consumption through premium request units (PRUs). Developers rarely considered the computational cost behind each interaction as pricing remained largely predictable. That abstraction has now disappeared.

GitHub Copilot's billing model measures AI usage directly through token consumption. Every input token, generated response, cached prompt, and model invocation contributes to enterprise AI spend. Organizations continue to pay their subscription fees, but AI consumption is now governed by AI credits, creating a variable-cost layer that engineering organizations must actively manage.



What Changed	Enterprise Impact
AI credits become the billing unit.	AI usage becomes directly measurable across the enterprise.
Credits are pooled across the organization.	Heavy users consume from a shared enterprise budget rather than individual allocations.
Premium reasoning models consume significantly more credits.	Model selection becomes a financial decision as well as a technical one.
Automatic fallback models are removed after credits are exhausted.	Exhausted AI credits trigger paid usage or access blocks. GitHub will not auto-switch to cheaper models.
A Copilot code review can incur two cost components: AI Credits for model usage and GitHub Actions minutes for its agentic review infrastructure. Treat this as a code-review-specific cost path, not a universal property of Copilot agents.	Organizations must govern two separate operational cost streams simultaneously.

This shift fundamentally changes AI governance.

Previously, engineering leaders focused on AI adoption. Today they must balance adoption with financial accountability, ensuring that developers receive the right AI capability without exposing the organization to uncontrolled operational expenditure.

A useful comparison is cloud computing. Infrastructure became economically sustainable only after enterprises introduced FinOps, observability, and workload optimization. Enterprise AI now requires the same level of operational discipline.

The Enterprise Implication

Subscription pricing encouraged organizations to ask: "How many developers should receive AI access?"

Token metering changes the question entirely to: "How should AI resources be governed across thousands of engineering decisions every day?"

The answer lies not in limiting AI usage, but in establishing governance policies that ensure every model invocation creates measurable business value.

The Model Cost Gap Is Real

Not every AI model costs the same. More importantly, not every engineering task requires the most capable reasoning model available.

The difference between economy and frontier models can exceed twenty times the cost for equivalent token volumes. When multiplied across thousands of developers and millions of interactions, habitual model selection becomes one of the largest drivers of enterprise AI expenditure.

Workload profile	Routing guidance	Control
Low complexity, high volume	Start with a lightweight model	Default policy
Routine engineering	Use a lightweight or versatile model	Compare acceptance and rework
Complex debugging and design	Use a versatile/powerful model selectively	Require value justification
Long-context or autonomous execution	Evaluate both model price and context volume	Set run/session budget
Regulated or high-risk change	Select on assurance and validation needs—not price alone	Human approval and audit trail

The lesson is straightforward: organizations should reserve premium reasoning models for high-value planning and decision-making rather than routine software implementation.

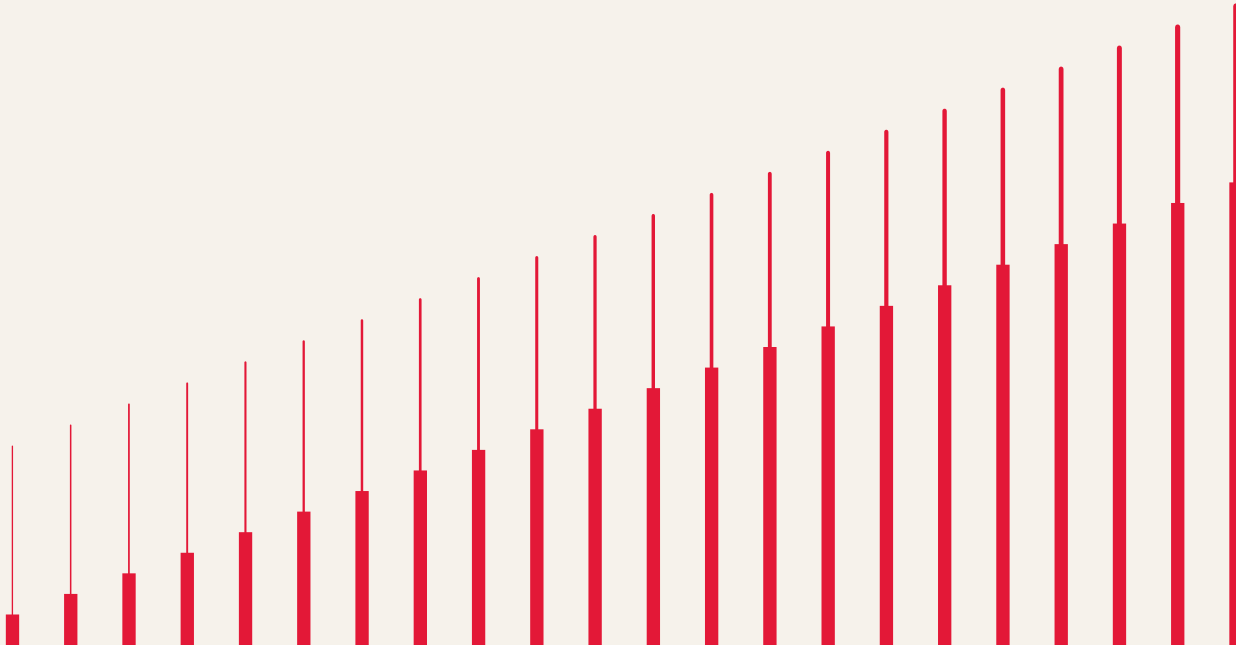
The Principle Every Engineering Team Should Adopt

Use the least-cost model that meets the task's assurance and quality needs: stronger reasoning for ambiguous design and complex diagnosis; lighter or mid-tier models for bounded execution, documentation, and routine refactoring. Validate outcomes with tests and human review. Use frontier reasoning models to define architecture, establish implementation strategy, produce technical specifications, and identify design trade-offs.
Then allow lower-cost models to execute well-defined implementation work repeatedly and consistently.

High-value reasoning should happen once. Routine execution should happen thousands of times at the lowest appropriate cost. Model routing, not developer restriction, is what produces sustainable AI economics.

Why This Matters at Enterprise Scale

A single premium interaction may appear inexpensive in isolation. Across thousands of developers, however, habitual use of the premium model rapidly compounds into millions of unnecessary tokens each month. Organizations that govern model selection through policy consistently achieve higher AI productivity while controlling operational expenditure. Those that allow unrestricted model selection typically experience the opposite: higher spend with little measurable improvement in delivery outcomes.



What Is Actually Consuming Your Tokens

One of the biggest misconceptions surrounding AI costs is that users primarily pay for the prompts they type. In reality, the visible prompt often represents only a small portion of the context processed during every model invocation.

Modern coding assistants continuously assemble additional information behind the scenes, including repository context, conversation history, instruction files, tool schemas, system prompts, and open files. Each of these elements contributes to token consumption.

Context Component	Typical Contribution
Repository and open file context	Highest
Conversation history	High
Instruction files and coding standards	Moderate
Tool schemas and MCP context	Moderate
System instructions	Continuous
User prompt	Often the smallest component

For many engineering teams, the majority of AI spend is therefore invisible during normal usage. Developers naturally focus on the prompt they type, while the platform silently processes substantially more contextual information.

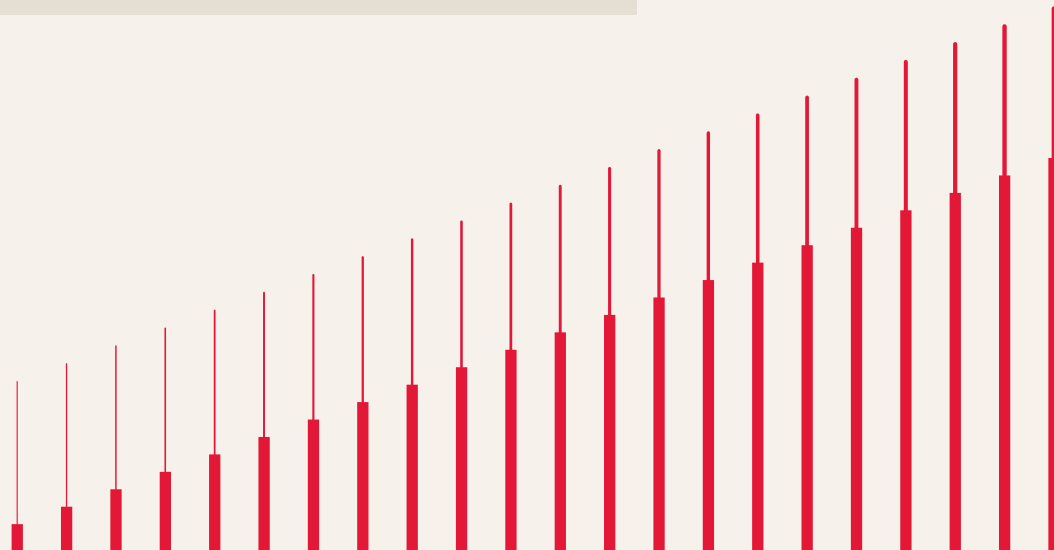
Two Hidden Drivers of Enterprise AI Spend

Context accumulation

Long-running conversations continually expand the amount of information sent to the model. As context grows, token consumption increases even if individual prompts remain short. Starting a fresh session after completing a logical unit of work often reduces unnecessary token usage while improving response quality.

Excessive workspace context

Leaving numerous files open, exposing unnecessary repositories, or loading excessive instruction sets causes every request to process information that may be irrelevant to the current task. Reducing unnecessary context frequently improves both cost efficiency and model performance.

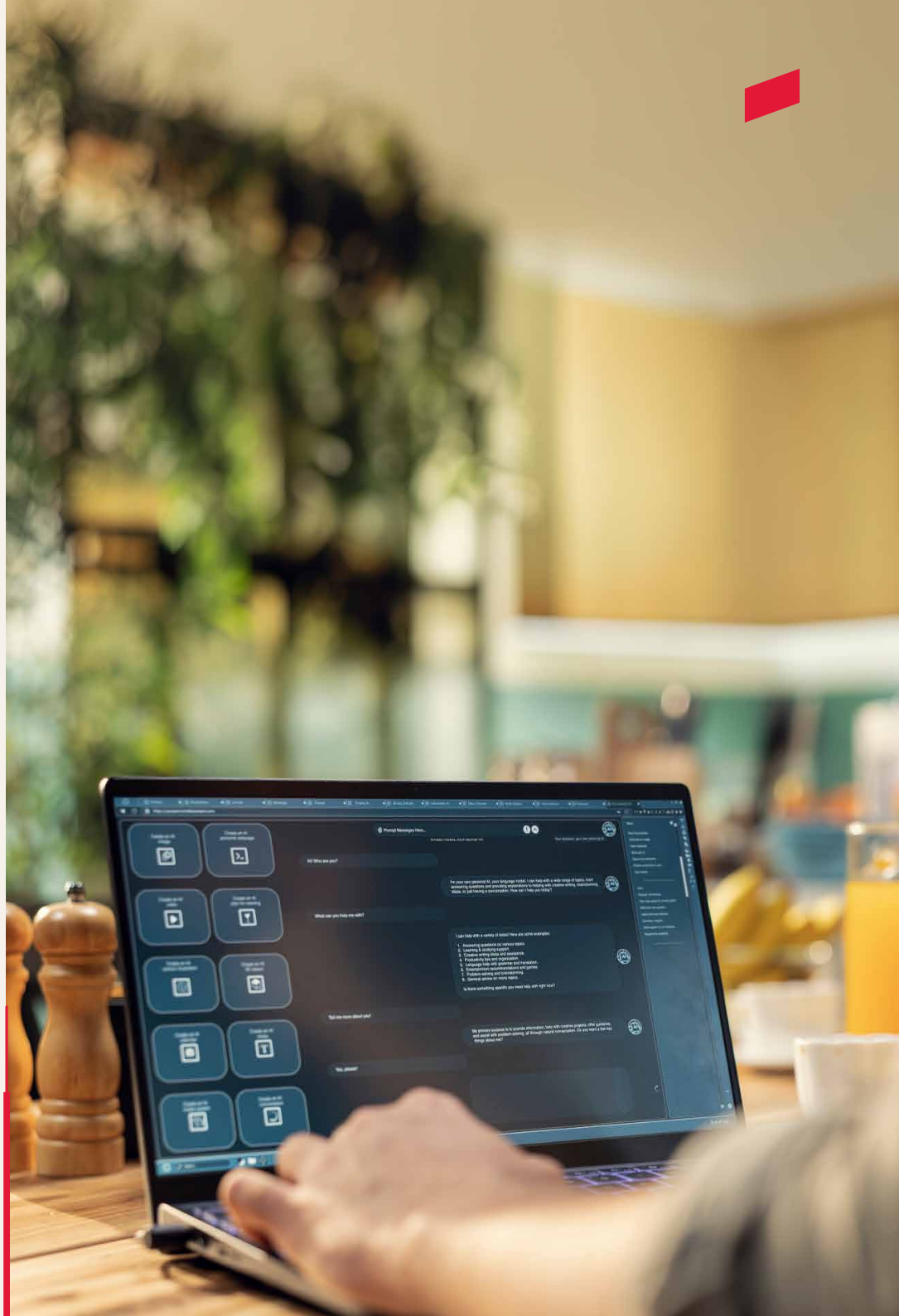


Prompt Engineering Alone Is Not the Answer

Well-written prompts certainly improve AI quality. However, prompt engineering addresses only a fraction of enterprise AI economics. The larger cost drivers include:

- Selecting the appropriate model.
- Controlling contextual scope.
- Governing session length.
- Routing work intelligently.
- Designing AI workflows.
- Applying FinOps policies across engineering teams.

Organizations that focus exclusively on prompt engineering optimize only one variable within a much larger operating model. Sustainable cost control requires governance across the entire AI development lifecycle.



Caching: The Lever Most Teams Get Wrong

Caching is frequently presented as a universal cost optimization technique. The reality is more nuanced. Different AI providers implement caching differently, and the financial impact depends heavily on workload characteristics. Enterprises should distinguish between two approaches:

Provider-level prompt caching

- Reuses identical prompt prefixes.
- Reduces repeated processing costs.
- Best suited for highly repetitive workflows.

Application-level semantic caching

- Reuses completed responses for semantically similar requests.
- Eliminates unnecessary model invocations.
- Often delivers greater savings at enterprise scale.

Caching should therefore be treated as an architectural design decision rather than a configuration setting. Evaluate any semantic-cache architecture separately for correctness, privacy, hit rate, and measured savings.

The 5 Layer FinOps Stack

Enterprise AI governance begins with visibility. Without accurate telemetry, organizations cannot understand where AI resources are consumed, which teams generate value, or which workflows require optimization.

The AI FinOps stack provides five foundational capabilities.

Layer	What to Build	Business Value
Observability	Centralized dashboards for AI credits, models, and workflow usage.	Makes AI consumption measurable across the enterprise.
Budget Management	Team Budgets, alerts, and spending thresholds.	Prevents uncontrolled cost escalation.
Model Governance	Enterprise model routing policies.	Matches capability with workload complexity.
Anomaly Detection	Identifies abnormal usage patterns.	Detects inefficient or excessive consumption early.
Quarterly AI finOps Reviews	Executive reviews of AI cost, productivity, and ROI.	Establishes continuous optimization as an operating discipline.

These capabilities should operate together. Visibility without governance produces reporting. Governance without visibility produces assumptions. Effective AI finOps requires both.

Model Routing Cheat Sheet

The objective of model routing is simple: Match the simplest model capable of completing the task.

Engineering Task	»	Recommended Model Tier
Documentation, explanations, code understanding.	»	Free
Boilerplate implementation.	»	Economy
Routine feature development.	»	Everyday
Architecture validation.	»	Mid
Enterprise system design.	»	Premium

Organizations should formalize routing through policy rather than relying on individual developer preference.

5 Routing Rules:

Default to free models whenever possible

Reserve premium models for planning and complex reasoning

Use Ask Mode before Agent Mode when code generation is unnecessary

Never launch autonomous agents without a complete technical specification

Monitor output size, since generated tokens typically cost substantially more than input tokens

The transition to token-metered AI development is not fundamentally about billing. It represents the emergence of AI as an operational resource that must be governed with the same rigor as cloud infrastructure, cybersecurity, and software delivery. Organizations that establish observability, model routing, AI finOps, and engineering governance now will be positioned to scale AI sustainably. Those that continue treating AI as an unrestricted productivity tool risk replacing one inefficiency with another.

The AI SDLC Operating Model: Pods, Sprints, and Metrics

AI has not simply accelerated software development. It has fundamentally redistributed where engineering effort is invested.

For decades, software delivery centered on writing code. Today, AI generates much of the implementation work, shifting the greatest demands toward planning, architecture, validation, governance, and quality assurance. Organizations that continue measuring productivity through traditional development metrics risk optimizing the wrong parts of the delivery lifecycle.

This shift requires more than adopting AI coding assistants. It requires redesigning the operating model that governs how engineering teams work.

The Shift Nobody Warned You About

The prevailing narrative around AI focuses on developer productivity. The reality is more nuanced. AI may change implementation, review, and validation workflows. Establish a baseline and evaluate effects locally through DORA outcomes, PR-flow measures, quality controls, and business outcomes; do not infer causality from usage telemetry alone. Requirements become more precise, architectural decisions become more consequential, and human review becomes more valuable than ever.

Recent industry research supports this pattern. AI-assisted development consistently improves code generation velocity while increasing pressure on downstream activities such as pull request reviews, testing, production validation, and incident management. Rather than eliminating engineering work, AI redistributes it across the software delivery lifecycle. This creates a new leadership challenge. The constraint is no longer developer throughput. The constraint is organizational capacity to validate, govern, and safely deploy AI-generated software.

The New Bottleneck

The scarcest engineering resource is no longer the ability to write code. It is the ability to evaluate it. Senior engineers, architects, quality engineering teams, and platform owners increasingly become the limiting factor in delivery velocity because they provide the judgment AI cannot replace.

Organizations that scale AI successfully invest as heavily in review capacity, testing discipline, and engineering governance as they do in developer productivity. Those that focus exclusively on code generation frequently discover that review queues, technical debt, and production defects expand at the same pace as AI output.

How Engineering Effort Changes

SDLC Phase	Traditional Delivery	AI-led Delivery
Requirements	Moderate	High
Architecture and Design	Moderate	Significantly Higher
Implementation	Highest	Significantly Lower
Testing	High	Moderately Lower
Deployment	Similar	Similar
Operations	Moderate	Higher

The implication is clear. AI reduces implementation effort but increases the value of engineering judgment throughout the remainder of the lifecycle. Organizations should therefore redesign delivery processes rather than expecting existing workflows to absorb exponentially higher development velocity.

Redesigning the Kanban Board for AI Pods

Traditional Kanban boards were designed for human developers progressing work sequentially through implementation. AI fundamentally changes that assumption.

An autonomous coding agent can generate an implementation within minutes. If planning, review, and validation remain unchanged, delivery bottlenecks simply move downstream. An AI-native workflow therefore introduces additional governance checkpoints before implementation begins.

Workflow Stage	Primary Owner	Exit Criteria
Backlog	Product Owner	Business value defined and prioritized.
Spec-Ready	Technical Lead	Requirements complete, bounded, testable, and approved.
Agent in Progress	AI Agent / Developer	Implementation generated within defined scope.
Human Review	Senior Engineer	Architecture, logic, and quality validated.
Test Gate	QA / SDET	Functional, regression, and security validation completed.
Done	Engineering Team	Acceptance criteria satisfied and deployment approved.

Why Spec-Ready Becomes the Most Valuable Stage

Spec-Ready is more than documentation—it is the control point that protects engineering quality, AI spending, and delivery predictability simultaneously. Well-defined specifications reduce unnecessary AI iterations, minimize ambiguity, lower review effort, and significantly reduce token consumption during autonomous execution. For engineering leadership, Spec-Ready establishes a measurable relationship between planning quality, AI cost, and delivery performance.

Sprint Ceremonies, Lightly Rewired

AI does not eliminate agile practices. It changes what those practices are intended to manage. Traditional ceremonies emphasized implementation progress. AI-native ceremonies prioritize planning quality, review capacity, engineering governance, and operational efficiency.

Ceremony	Traditional Focus	AI-Native Focus
Sprint Planning	Estimate development effort	Estimate specification, review, testing, and AI execution effort
Daily Standup	Individual implementation progress	AI output, review bottlenecks, specification quality, and blocked work
Sprint Review	Demonstrate completed functionality	Demonstrate business outcomes, AI utilization, and delivery metrics
Retrospective	Process improvements	Model selection, AI quality, governance effectiveness, and cost optimization
Backlog Refinement	Prioritize future work	Produce implementation-ready technical specifications

Agile remains relevant. The difference is that planning and validation become the dominant activities rather than implementation itself.

Spec-Ready: The Gate That Protects Your Budget

Organizations often assume AI increases productivity simply by writing code faster. In practice, AI productivity begins much earlier. A specification that lacks clarity, scope boundaries, architectural guidance, or measurable acceptance criteria forces autonomous agents into repeated clarification cycles. Those iterations consume additional AI credits, increase review effort, and frequently produce inconsistent implementations. Spec-Ready establishes a governance gate before autonomous execution begins.

Every AI Story Should Satisfy These Conditions

Requirement	Purpose
Business objective clearly defined	Align implementation with intended outcome.
Explicit scope boundaries.	Prevent uncontrolled AI expansion.
Testable acceptance criteria.	Enable objective validation.
Affected systems identified.	Limit unnecessary repository context.
Technical constraints documented.	Protect performance, security, and compliance.
Implementation plan approved.	Guide autonomous execution.
Model and execution strategy selected.	Match reasoning capability with workload.
Estimated AI credit budget assigned.	Improve financial predictability.
Technical Lead approval completed.	Authorize autonomous implementation.

Stories that fail these conditions should remain outside AI execution until they satisfy every governance requirement.

Spec-Ready Is an Enterprise Control Mechanism

Organizations often assume AI increases productivity simply by writing code faster. In practice, AI productivity begins much earlier. A specification that lacks clarity, scope boundaries, architectural guidance, or measurable acceptance criteria forces autonomous agents into repeated clarification cycles. Those iterations consume additional AI credits, increase review effort, and frequently produce inconsistent implementations. Spec-Ready establishes a governance gate before autonomous execution begins.

The Metrics Stack: What to Measure Every Sprint

Traditional software metrics remain important. However, they no longer provide a complete picture of engineering performance. AI-assisted delivery introduces new operational variables that leadership teams must measure alongside conventional software engineering metrics. A balanced measurement framework combines delivery, productivity, financial, and governance indicators.

Delivery Metrics

- Deployment frequency
- Change lead time
- Pull request review duration
- Change fail rate
- Deployment rework rate

Financial Metrics

- Total AI credit cost
- AI cost per active user
- AI cost per merged PR
- AI cost per deployed change

AI Productivity Metrics

- Agent adoption rate
- Suggestion acceptance rate
- Line acceptance rate
- AI-attributed LoC changes
- Human-review effort (determines whether coding gains are offset by review burden)

Governance Metrics

- Human-review compliance rate: Confirms that applicable changes receive required human oversight.
- Security control pass rate: Measures successful SAST, secret, dependency and IaC validation.
- AI policy-compliance rate: Tracks adherence to approved models, data and agent policies.
- AI audit-trace completeness: Confirms required prompts, models, approvals and actions are traceable.

Traditional software metrics remain important. However, they no longer provide a complete picture of engineering performance. AI-assisted delivery introduces new operational variables that leadership teams must measure alongside conventional software engineering metrics. A balanced measurement framework combines delivery, productivity, financial, and governance indicators.

Leadership Takeaway

The organizations creating lasting competitive advantage from AI are not those generating the most code, but those redesigning software delivery around planning, governance, validation, and measurable operational outcomes. AI changes where engineering effort creates value. The operating model must evolve accordingly.

AI-assisted software engineering represents the most significant change to enterprise delivery models since agile and DevOps. Yet the objective remains unchanged: Deliver high-quality software predictably, securely, and at scale.

The difference is that AI has shifted the economics of software development. Mechanical implementation is increasingly automated, while planning, architecture, governance, and engineering judgment become the defining capabilities of high-performing organizations.

Enterprises that redesign their SDLC around these principles will realize sustained gains in productivity, quality, and cost efficiency. Those that simply add AI to existing workflows will accelerate code generation without fundamentally improving software delivery.

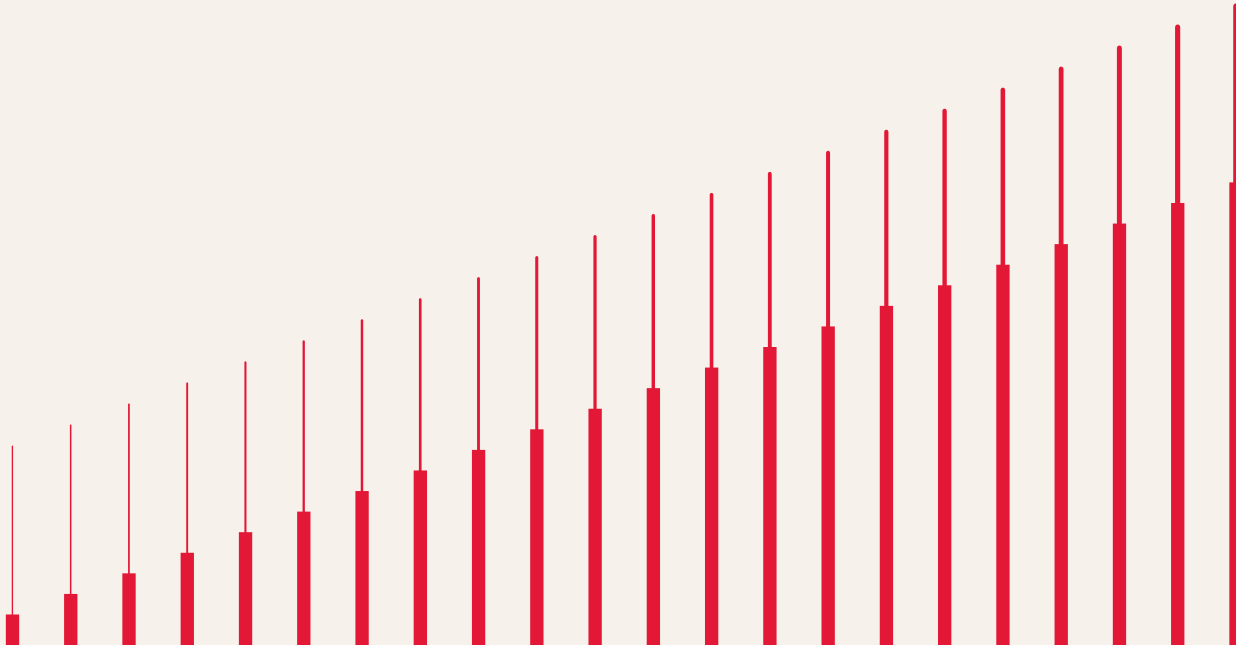
Governing Enterprise AI at Scale

Moving AI from isolated developer experimentation to enterprise-wide adoption requires more than enabling new tools. It demands governance that protects intellectual property, customer data, software quality, regulatory compliance, and financial accountability while allowing engineering teams to innovate at speed.

The organizations achieving sustainable AI transformation are not those imposing the most restrictive controls. They are the ones embedding governance directly into software delivery so that secure, compliant, and financially responsible AI usage becomes the default engineering experience.

Secure AI Development Begins Before Code Is Merged

AI can generate production-ready code within minutes. That same speed also increases the risk of introducing vulnerabilities, insecure dependencies, exposed credentials, or licensing issues into enterprise applications. Security therefore cannot remain a downstream activity. It must become an automated checkpoint throughout the AI-assisted development lifecycle.



SAST Pre-Merge Gating

Every AI-generated pull request should pass automated validation before human review begins.

Control	Purpose
Static Application Security Testing (SAST)	Detect vulnerabilities before merge.
Secret Scanning	Prevent credential exposure.
Software Composition Analysis (SCA)	Validate open-source dependencies and licensing.
Infrastructure as Code Scanning	Secure cloud deployment configurations.
AI-Generated Code Review Policies	Ensure generated code meets enterprise standards.

By automating these controls, organizations reduce manual review effort while preventing security issues from progressing through the delivery pipeline.

Security as a Quality Gate

Security should no longer be treated as a compliance checkpoint performed after development. Instead, it becomes part of the Definition of Done (DoD) for every AI-assisted implementation. This approach allows AI to accelerate delivery without compromising enterprise security posture.

Data Privacy, Intellectual Property, and Agent Governance

Enterprise AI systems operate on information that is often commercially sensitive, regulated, or proprietary. Without appropriate governance, organizations risk exposing customer data, intellectual property, confidential source code, or regulated information through AI interactions.

Responsible AI adoption therefore requires governance across both technology platforms and engineering practices.

Governance Framework

Governance Area	Enterprise Objective
Data Privacy	Protect regulated and sensitive information.
Intellectual Property	Prevent unintended disclosure of proprietary assets.
Agent Governance	Define permissible autonomous behavior.
Prompt Governance	Standardize prompt creation, review, reuse, and sensitive data handling.
Audit Logging	Maintain complete records of AI interactions, model usage, approvals, and governance decisions.
Regulatory Compliance	Align AI operations with enterprise compliance obligations.

Prompt Governance

Prompt engineering should evolve into an enterprise governance capability rather than remaining an individual developer skill. Organizations should establish policies covering approved prompt templates, prohibited data categories, confidential information handling, reusable engineering instructions, prompt version management, and secure prompt repositories.

Treating prompts as governed engineering assets improves consistency, protects sensitive information, and enables repeatable AI outcomes across development teams.

Audit Logging

Enterprise AI governance depends upon transparency. Every significant AI interaction should be traceable throughout the software delivery lifecycle.

Audit records should include model selection, prompt execution, generated outputs, approval decisions, deployment history, security validation, and policy exceptions. Comprehensive audit logging strengthens compliance while providing engineering leadership with the operational visibility required to improve AI governance continuously.

Governance Is an Engineering Capability

Organizations often view governance as slowing innovation. The opposite is increasingly true. When governance becomes a part of everyday engineering workflows, developers spend less time seeking approvals, security teams identify fewer exceptions, and leadership gains greater confidence in scaling AI across the enterprise. Governance therefore becomes an accelerator rather than a constraint.

The 90-Day Enterprise AI Adoption Plan

Successful enterprise AI programs rarely begin with large-scale transformation initiatives. They begin with operational discipline. The first 90 days should establish the governance foundations that enable AI to scale predictably across engineering organizations.

Phase 1

Establish Visibility

- Baseline current AI consumption.
- Enable enterprise observability.
- Measure AI credits, model utilization, and engineering usage patterns.
- Identify high-value pilot teams.

Phase 2

Build Governance

- Publish model routing policies.
- Define prompt governance standards.
- Implement spending thresholds.
- Introduce AI finOps reporting.

Phase 3

Redesign Engineering Workflows

- Introduce Spec-Ready planning.
- Update agile ceremonies.
- Redesign AI delivery pods.
- Integrate AI governance into software delivery.

Phase 4

Strengthen Engineering Controls

- Automate security validation.
- Implement audit logging.
- Standardize approval workflows.
- Expand governance reporting.

Phase 5

Measure Business Outcomes

Monitor:

- Delivery productivity.
- AI utilization.
- Engineering quality.
- Financial performance.
- Governance maturity.
- Business impact.

Phase 6

Scale Continuously

Expand successful practices across engineering organizations while refining governance using operational telemetry and business outcomes. Enterprise AI maturity is achieved through continuous optimization rather than one-time implementation.

Partner with Tech Mahindra

Enterprise AI transformation extends far beyond technology deployment. Organizations must redesign engineering practices, establish governance frameworks, implement AI finOps, modernize software delivery, and continuously measure business outcomes. Tech Mahindra helps enterprises build these capabilities through an integrated transformation approach.

Enterprise AI Transformation Framework

Phase	Tech Mahindra Capability
Assess	AI readiness, engineering maturity, governance assessment
Design	AI operating model, SDLC redesign, AI finOps framework
Implement	Platform deployment, developer enablement, governance integration
Optimize	Continuous measurement, productivity improvement, operational governance

Rather than delivering isolated AI implementations, Tech Mahindra helps organizations establish repeatable operating models that enable AI to scale responsibly across the enterprise while balancing innovation, governance, financial accountability, and engineering productivity.

A Note on Partnership

As enterprises move from experimentation to enterprise-wide AI adoption, the challenge is no longer selecting the right models or tools. It is building an operating model that enables AI to deliver measurable business value consistently, securely, and at scale.

Tech Mahindra partners with organizations throughout this journey, from assessing AI readiness and modernizing engineering practices to implementing governance, AI finOps, and enterprise-scale delivery models. By combining deep engineering expertise with proven transformation frameworks, Tech Mahindra helps organizations govern AI responsibly while accelerating innovation and realizing sustainable business outcomes.

The Final Word

The success of enterprise AI will not be determined by how many copilots an organization deploys or how many lines of code AI generates. It will be determined by how effectively organizations govern AI as an operational capability.

Sustainable AI transformation depends on disciplined engineering practices, transparent governance, responsible financial management, and measurable business outcomes working together. Organizations that establish these foundations today will be better positioned to scale AI confidently as technologies, models, and development practices continue to evolve.

AI adoption is only the beginning. Long-term value comes from building an operating model that allows innovation, governance, and business performance to advance together.

About the Author

Srini Burra is Vice President and Head of the North America Consumer Products business at Tech Mahindra, where he partners with leading CPG companies on growth, technology modernization, and AI-led transformation. His work spans consumer goods, manufacturing, digital commerce, and enterprise platforms, with a consistent focus on turning technology investment into measurable business outcomes.

His current focus is how AI is reshaping the software development lifecycle from requirements and code generation through testing, governance, and the operating models needed to sustain them at enterprise scale.

Srini Burra
Vice President and Head of North America Consumer Products Business, Tech Mahindra

References

1. Gartner. (2025, June 25). Gartner predicts over 40 percent of agentic AI projects will be canceled by the end of 2027. Gartner Newsroom. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

About **Tech Mahindra**

Tech Mahindra (NSE: TECHM, BSE: 532755) offers technology consulting and digital solutions to global enterprises across industries, enabling transformative scale at unparalleled speed. With 146,000+ professionals across 90 countries, Tech Mahindra provides a full spectrum of services including consulting, information technology, enterprise applications, business process services, engineering services, network services, customer experience & design, AI & analytics, and cloud & infrastructure services. It is the first Indian company in the world to have been awarded the Sustainable Markets Initiative's Terra Carta Seal, which recognizes global companies that are actively leading the charge to create a climate and nature-positive future. Tech Mahindra is part of the Mahindra Group, founded in 1945, one of the largest and most admired multinational federation of companies.

TECH mahindra

www.techmahindra.com
www.twitter.com/tech_mahindra
www.linkedin.com/company/tech-mahindra