Powering next-generation GCC automation

May 2026

Global capability centres (GCCs) now anchor global delivery at scale. **Dheeraj Gangrade**, **Himadri Ganguly**, and **Rahul Dutta** examine how Java—already core to most GCC enterprise systems—is positioned to power the full spectrum of next-generation automation across the technology stack.

Contents

01

When autonomy meets enterprise reality

Picture this: A GCC runs with engineering and R&D (ER&D) teams spread across multiple countries, time zones, and cloud environments. Workstreams are interconnected, but the moment complexity scales, communication breaks down between the nodes, tasks aren't mapped to the right resources, and workflows fall out of sync.

Now imagine a Java-based automation layer woven into this fabric.

Agents communicate reliably across distributed networks through Java's robust, asynchronous messaging libraries. When a task needs the right expertise, Java-powered frameworks dynamically discover and engage the most capable agent based on real-time context and availability. Deep cloud-native compatibility allows these agents to be deployed and scaled exactly where the data lives, potentially reducing latency and enabling real-time collaboration across continents. And Java's concurrency and reactive libraries enable agents to manage parallel workflows and respond to events efficiently.

The result? Resources are optimally allocated. Errors are caught autonomously. Experiment turnaround times are dramatically reduced. Testing, iteration, and innovation cycles are accelerated through intelligent, self-orchestrating systems built on a language the enterprise already trusts.

As a widely adopted enterprise-grade platform built to support scalability, reliability, and long-running systems, Java can serve as a solid foundation on which agentic AI solutions may be effectively designed and deployed. Its strong concurrency models, mature ecosystem, and integration with cloud-native and distributed architectures make it a practical option for orchestrating autonomous agents that monitor complex processes, process large volumes of data, and make informed decisions in real time. As a result, ER&D wings can move from reactive oversight to proactive, intelligent experimentation, thus potentially unlocking faster innovation cycles and more confident decision-making at scale.

02

Redefining distributed work for enterprise impact

In a year marked by geopolitical uncertainty and rapid advances in AI, GCCs are quietly reshaping how global enterprises operate. Even as organisations rethink supply chains and operating models, GCCs are expanding their mandates, shifting from execution engines to innovative intelligence hubs that design, build, and run globally distributed systems. India's GCC ecosystem—anchored in engineering talent, digital maturity, and renewed policy confidence—is increasingly where complex enterprise workflows are orchestrated and innovation is scaled. As automation gives way to autonomy, GCCs are emerging as the proving ground for the next frontier of enterprise AI, buoyed by policy tailwinds.

The 2026 Union Budget's changes to the safe harbour regime is expected to unlock a wave of investments into the country by GCCs or the R&D arms of multinationals.[1] The proposal to consolidate all IT services, including IT-enabled services, software development services, knowledge process outsourcing (KPO), and contract R&D services under a single category of 'Information Technology Services'[2] makes tax compliance more predictable for the technology sector. For GCCs in India, this marks a meaningful reprieve from long-standing tax uncertainty and classification disputes.[3] This is also expected to streamline compliance processes and decrease disputes over transfer pricing—an area that has long been a bone of contention for GCCs in India, creating uncertainty that has constrained long-term planning and scale, particularly for large GCCs.[4]

These developments are likely to lead to increased capital inflows, technology transfers, and enhanced innovation activities within the country, strengthening India's position as a preferred destination for distributed operations. In recent years, GCCs in India have made significant progress as cost-effective innovators and multifunctional excellence hubs, evolving into drivers of front-end value for their headquarters.[5]

1 The Economic Times, Budget 2026: Tech, GCCs set sail with safe harbour shield

2 PIB press release

3 CNBC TV18, Global Capability Centres get tax certainty in Budget 2026; safe harbour margin at 15.5%

4 The Economic Times, Budget 2026: Tech, GCCs set sail with safe harbour shield

5 PwC, Catalysing value creation in Indian global capability centres

PwC India's previous research found that 92% of product-based GCCs and 88% of service-based GCCs in India offer advanced technology services, including innovation labs, product development and design, prototyping, data analytics, AI, and machine learning projects.[6] This depth of capability explains why India's GCCs have emerged as central to distributed workflows:

Lower costs: Establishing functions in GCCs located in countries with lower labour and operational expenses significantly reduces companies' costs related to hiring, infrastructure, and logistics.

Access to global talent: A distributed model through GCCs enables global firms to tap into unique expertise and rich talent pools across diverse regions.

Mitigating supply chain risks: Distributing operations reduces dependency on a single location and mitigates risks from supply chain disruptions. A global technology giant is a case in point. It is diversifying its supply chain and operational functions, including manufacturing, to limit dependency on particular regions and offset potential tariff-related impacts.

Supporting sustainability goals: A distributed model allows companies to leverage region-specific expertise to foster green innovations. Locating teams nearer to regional manufacturers facilitates designing products with locally sourced materials, thus reducing emissions, shrinking logistics footprint, and minimising long-distance transport needs.

With GCCs firmly embedded as strategic anchors of global delivery, attention is now turning to how their impact can be amplified further.

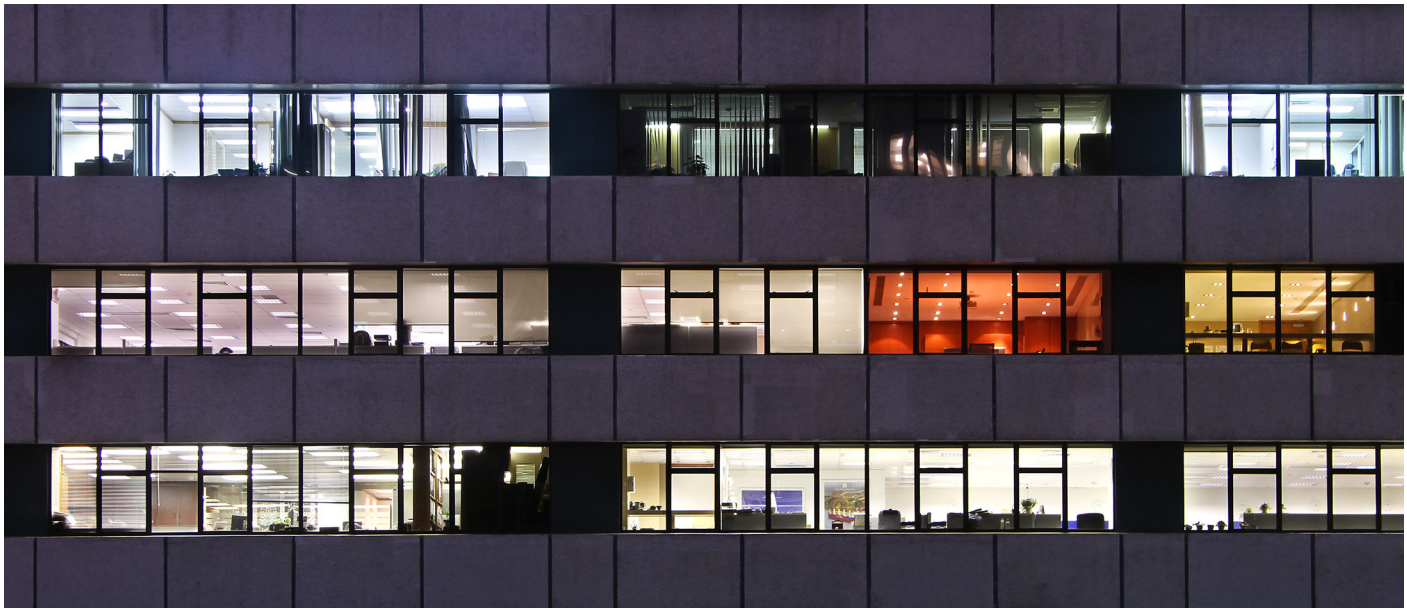
Our take

03

Powering smarter automation for GCCs

Our experience indicates that for most enterprises and GCCs, mission-critical systems—ranging from ERP integrations and cloud-native microservices to security architectures and compliance frameworks—are already deeply rooted in Java. Despite languages such as Python and C++ dominating GPU-accelerated and frontier AI workloads, Java's relevance has not diminished. Instead, its ecosystem has continued to evolve, offering extensive libraries, robust scalability, and, importantly, the ability to integrate seamlessly with the enterprise systems that organisations already depend on.

Java's standing therefore persists where it matters most for GCCs—reliability, scale, security, and enterprise integration. Rather than competing head-on for raw GPU performance, Java continues to anchor AI-enabled systems at the enterprise layer, reinforcing its position as a robust, enterprise-first language even as AI landscapes shift. Its evolving ecosystem, modern Java virtual machine (JVM) features, and strong tooling sustain its critical role in large-scale, mission-critical applications, particularly in enterprise, big data, and cloud systems.



Java's strengths in enterprise environments

Robust, scalable platform

Java offers a mature and stable environment designed to efficiently handle the concurrent processing and orchestration demands of AI-enabled enterprise systems distributed across GCC operations.

1

Seamless integration

Java's extensive ecosystem and interoperability help GCCs to smoothly integrate AI models and agent frameworks with existing enterprise applications and cloud platforms.

2

Security and compliance

As data privacy and regulatory requirements become paramount, Java's built-in security features and continuous updates help GCCs protect sensitive data while deploying AI solutions responsibly.

3

Cloud-native and multitenant architecture support

Java simplifies building scalable, cloud-native applications essential for flexible, cost-efficient AI service deployment across global GCC operations.

4

Cross-platform portability and scalability

Agentic AI systems on Java can operate seamlessly across diverse operating systems and hardware common in large enterprises. Its mature JVM optimises resource management and supports scalable deployments. Java's vast libraries, frameworks, and APIs allow reliable integration with heterogeneous enterprise systems and third-party services, simplifying maintenance and scalability.

5

Enhanced native interoperability

The Project Panama initiative allows Java-based agentic AI systems to leverage optimised native libraries and hardware acceleration, boosting performance for computation-intensive GCC workflows.

6

While agentic AI represents the frontier of intelligent automation, Java's value to GCCs spans the entire transformation lifecycle. What follows illustrates how Java powers full-spectrum automation:

Platform modernisation: Java can enable GCCs to break down monolithic architectures into modular, microservices-based systems that support independent deployments, team autonomy, and rapid iterative development.

Cloud adoption: Java simplifies the development of scalable, multitenant, cloud-native applications through enterprise-grade software development kits (SDKs) that abstract tenant handling, outbound communication, and resilience mechanisms.

Agentic AI adoption: Java's strong concurrency models, reactive libraries, and seamless integration with enterprise systems make it the natural orchestration engine for deploying autonomous AI agents that sense, plan, execute, and learn.

Security and compliance: Java's built-in security features—including authentication, encryption, multitenant isolation, and continuous patching—ensure that GCCs can protect sensitive data and meet evolving regulatory mandates without compromising agility.

Observability and monitoring: Java's mature observability ecosystem—encompassing real-time telemetry, integrated debugging tools, and structured monitoring—provides end-to-end visibility across distributed services and automated workflows.

04

Scalable solutions at work

As organisations accelerate platform modernisation and cloud adoption, Java is often the technology of choice due to its proven maturity and scalability. The following case studies illustrate how Java can power the full spectrum of GCC automation—spanning seamless cloud integration, intelligent workflow orchestration, enterprise-grade security, and reliable, high-performance execution at scale.

A. How a large ERP vendor's GCC reimagined its ecosystem

Background

The ER&D GCC of this German enterprise faced challenges across its ecosystem—from integration complexity and scalability bottlenecks to manual testing of inefficiencies and fragmented workforce processes. We partnered with the GCC to architect and deliver Java-based solutions that transformed its operations from fragmented and manual processes to integrated, automated, and resilient workflows.

Challenge

Enterprise applications struggled to reliably integrate with systems. Complex open data control/OpenAPI services, authentication handling, tenant isolation, and frequent outbound communication failures created instability and unpredictable system behaviour across distributed environments.

Our solution

We built applications that integrated seamlessly with the client's services. Java enabled secure connectivity, authentication, multitenant routing, and resilience features including caching, retries, circuit breakers, timeouts, and rate limiting, ensuring stable, predictable communication across the enterprise landscape.

Java features utilised: Secure connectivity, authentication, multitenancy, caching, retries, circuit breakers, timeouts, rate limiting, resilient communication APIs

Challenge

The company required cloud-native applications capable of supporting multiple tenants, scaling on demand, and maintaining resilience across environments. Building this from scratch significantly increased development time, cost, and risk.

Our solution

Our experts used Java and a cloud software development kit (SDK) to build multitenant SaaS applications running efficiently on a leading cloud business technology platform. The SDK abstracted tenant handling, outbound requests, and resilience mechanisms, allowing development teams to focus on business logic while delivering scalable, secure cloud solutions with faster time-to-market.

Java features utilised: Cloud SDK support, multitenant abstractions, scalability, security, faster development with SDK

Challenge

Testing was a critical bottleneck for integrated applications. Manual test creation, inconsistent execution, slow feedback cycles, and lack of automation impacted quality, delayed releases, and hindered continuous delivery.

Our solution

We built the Future of Testing (FOT) platform using Java with Spring Boot microservices. This automation platform orchestrates test generation, execution, and reporting through event-driven and asynchronous workflows, significantly improving test accuracy, reducing manual effort, accelerating delivery cycles, and enabling continuous integration/continuous delivery practices.

Java features utilised: Spring Boot microservices, event-driven programming, asynchronous workflows, and concurrent processing

Challenge

Enterprise workflows spanned multiple systems. Orchestrating these reliably was complex, error-prone, and susceptible to failures and latency issues.

Our solution

Our experts leveraged Java's robust ecosystem to build modular, event-driven architectures capable of orchestrating complex workflows across ERP and non-ERP components. Java services communicated seamlessly with databases, external APIs, and cross-language microservices, enabling reliable automation and smooth data exchange across the enterprise.

Java features utilised: Modular design, event-driven architectures, APIs integration, cross-language microservices communication

Challenge

The organisation faced significant gaps in recruitment, onboarding, performance management, and talent development. These inefficiencies reduced team productivity and directly impacted R&D effectiveness and business outcomes.

Our solution

We built Java-based integrations with a cloud-based HR software to automate and streamline HR workflows, including hiring, performance reviews, document management, and employee lifecycle processes. This ensured that business and R&D teams had timely access to skilled resources while HR processes ran smoothly across the enterprise.

Java features utilised: Enterprise integration, automation via Java backend services for cloud-based HR systems

B. How an online marketplace was able to solve business challenges across its entire technology stack

Background

The India GCC of the online marketplace plays a critical role in supporting complex global operations that underpin the company's global fulfilment and technology innovation. These activities involve developing and maintaining thousands of microservices that power customer-facing applications and backend automation.

The company's earlier monolithic architecture hindered scalability, delayed independent deployments, and restricted team autonomy—factors that slowed innovation and increased risks of failure. By adopting Java microservices, the organisation transformed its operational environment into a scalable, modular system with independent services. Java's strong concurrency support and mature libraries enabled reliable communication between services, dedicated data ownership per service, and rapid iterative development, fuelling continuous innovation and operational excellence within the teams.

Challenge

Customer-facing workloads demanded hundreds of parallel service calls per request. Failures in integration, increased latency, and timeouts directly impacted not only customer experience but also the quality and reliability of outputs at massive scale.

Our solution

Java's ability to efficiently handle multiple simultaneous tasks, along with built-in safeguards such as automatic retries, timeouts, and fallback mechanisms, ensured that critical systems remained fast and reliable even under heavy load. Integrated monitoring tools provided real-time visibility for debugging and maintaining system stability across distributed services.

Java features utilised: Concurrency (multithreading), automatic retries, timeouts, fallback mechanisms, monitoring and debugging tools

Challenge

Coordinating vast fleets of robots within automated workflows required real-time decision-making, fault tolerance, and safe concurrency. Failures could disrupt throughput and delay delivery of key projects and operational SLAs.

Our solution

Java-based orchestration systems used reliable concurrency models, event-driven workflows, and robust telemetry to efficiently manage robotic fleets at scale. This ensured predictable, safe operations across complex automation systems.

Java features utilised: Concurrency models, event-driven workflows, telemetry, and monitoring for fault tolerance and reliability

Challenge

Managing thousands of runtimes, patches, and cryptographic libraries critical to infrastructure posed significant complexity and security risks.

Our solution

Unified Java distribution ensured consistent security, timely patches, and optimised performance. Java's mature observability tools simplify compliance and monitoring across global infrastructure, reducing risk and operational overhead.

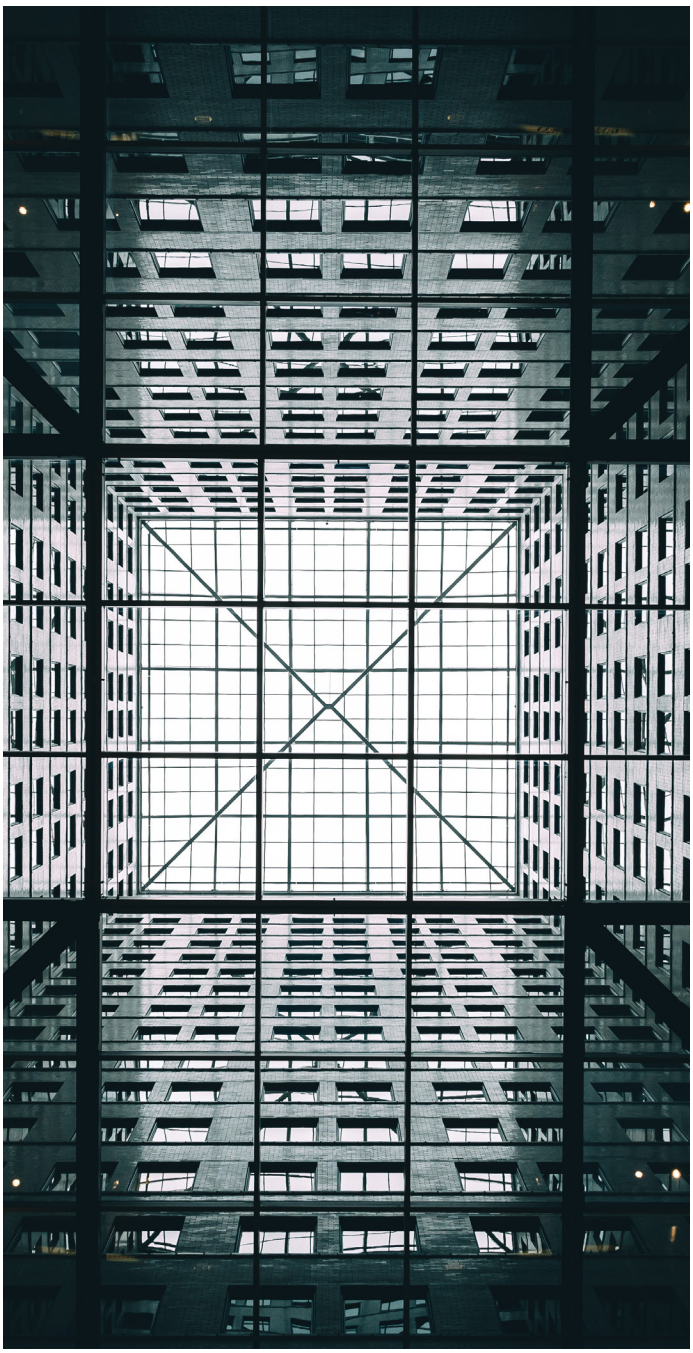
Java features utilised: Unified Java runtime distribution, security features, observability and monitoring tools for compliance and risk reduction

05

Select recommendations

As discussed in the earlier sections, Java is well-positioned to address the complex demands of GCC environments due to its inherent platform portability, extensive and mature ecosystem of libraries and frameworks, and robust support for cloud-native architectures. These strengths support the deployment and scaling of intelligent agents across distributed networks and heterogeneous systems typical of GCC operations. By leveraging Java's proven reliability and versatility, organisations can build resilient, interoperable AI solutions that drive innovation, enhance collaboration, and efficiently manage resources in a globally distributed enterprise landscape.

In view of this, GCCs could use Java to progress beyond task-level automation towards autonomous, intelligence-driven operations across core enterprise functions. In practice, GCCs could leverage Java-based AI agents to:



1

Modernise ERP ecosystem management

Deploy Java-based AI agents that continuously monitor ERP landscapes, anticipate system issues, and trigger corrective actions before disruptions occur. By using digital twins of enterprise systems, these agents can safely simulate upgrades and configuration changes—enabling risk-free optimisation of mission-critical platforms.

2

Orchestrate marketplace fulfilment in real time

Implement autonomous AI agents that dynamically coordinate inventory, logistics, and fulfilment workflows across distributed systems. Built on Java's enterprise-grade scalability, these agents respond in real time to variables such as demand spikes, weather conditions, and traffic disruptions—improving delivery speeds, lowering costs, and reducing environmental impact.

3

Reimagine HR workflows

Adopt agentic AI to proactively analyse workforce trends, identify emerging skill gaps, and recommend personalised reskilling pathways. Java-based agents can also autonomously source and screen talent across job portals and professional networks, accelerating hiring while improving quality of fit.

4

Accelerate enterprise innovation cycles

Use Java-enabled AI agents to automate experimental design, generate hypotheses, and synthesise insights from internal data and external knowledge sources. This allows GCC teams to compress innovation timelines, run parallel experiments at scale, and translate insights into production systems faster.

5

Strengthen governance, risk, and compliance

Deploy AI agents that continuously monitor compliance across enterprise systems, predict regulatory risks, and execute automated remediation actions. Leveraging Java's strengths in security and transactional integrity, these systems can adapt as regulations evolve, reducing manual oversight and compliance exposure.

AI-driven automation is fast becoming central to improving time-to-market and organisational agility, the two strategic imperatives for GCCs. By automating complex tasks such as experiment orchestration, data harmonisation, and semantic reasoning, Java-based agentic AI can enable faster iteration cycles and more confident decision-making. This, in turn, allows globally distributed teams to respond rapidly to the changing business environment, strengthening both innovation velocity and operational resilience.

Source: PwC analysis

Reimagining GCC project delivery with an agentic marketplace

GCCs today are faced with an ever-growing complexity of tools, data silos, and fragmented workflows spanning distributed teams and geographies. The challenge is no longer merely automating discrete tasks but creating intelligent orchestration—a unified platform that delivers context-aware, collaborative, and secure project execution at enterprise scale.

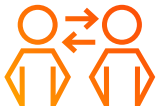
To address this, we have developed an agentic marketplace powered by Java. It is a centralised, modular ecosystem of AI-powered agents, each designed to automate and enhance distinct facets of the project lifecycle. Unlike traditional monolithic platforms, this marketplace offers a flexible, composable architecture where each agent acts as a reusable, configurable asset that plugs into a shared digital knowledge hub. GCCs can unlock several benefits from the agentic marketplace:

Unified institutional memory

Rather than scattering project knowledge across disconnected tools or relying on individuals' memory, the marketplace consolidates all contextual data into a single, governed source of truth—accessible in real time by every stakeholder, from programme managers to developers across GCC locations.

Accelerated onboarding and decision-making

New team members or project leads rapidly come up to speed by querying "project memory" through the agents that surface relevant historical context, past decisions, and actionable insights instantly.



Seamless cross-tool orchestration

Through standardised protocols, agents integrate deeply across platforms and collaboration environments—enabling smooth transitions and synchronised updates without manual intervention or disruptive tool switching.

Context-rich developer experience

The platform integrates directly with coding environments, allowing seamless handover of structured sprint data to Integrated Development Environments (IDEs).

Enterprise-grade security and compliance

The agentic marketplace ensures project data remains securely isolated within a governed architecture, supporting compliance mandates while enabling agile, distributed delivery.

Strategic advantage

By centralising institutional knowledge and enabling standardised yet flexible workflows, GCCs can dramatically improve delivery consistency, enhance governance across distributed teams, and accelerate innovation within complex, multi-client portfolios.

This paradigm fundamentally transforms how GCCs approach project delivery. By turning fragmented processes and siloed data into an intelligent, interconnected ecosystem, the agentic marketplace can unlock compounding institutional intelligence and empower GCC teams to deliver faster, smarter, at scale, and more securely.

06

Looking ahead

The future GCC operating model is poised to undergo a decisive shift—from a linear, effort-based construct to an agent-led, capacity-on-demand paradigm, where AI agents become the primary unit of delivery across augmentation, managed services, and transformation programmes.

This agentic transformation within GCCs—anchored predominantly on Java as the enterprise platform—will enable scale, resilience, and reuse.

By industrialising value propositions into reusable agentic assets, orchestrated through a governed marketplace, execution will be fundamentally re-wired across the lifecycle—from solutioning and build to run and optimisation.

The impact will be structural: dependence on linear FTE scaling will reduce by 25–35%,* while effective delivery capacity will expand by 1.5–2.0x through agent augmentation. Core engineering activities across integration, application development, microservices, and data will see 30–50% cycle-time compression, 20–40% productivity uplift, and 15–25% lower defect leakage, driven by standardised, always-on agent execution.

Commercially, this would entail a shift from traditional T&M and managed-capacity models toward asset-backed, outcome-linked pricing. The result would be a 5–10 percentage-point improvement in gross margins, alongside 20–30% cost optimisation for clients and a ~40% reduction in time-to-value. For GCCs, this would translate into faster program execution, higher throughput, and predictable delivery outcomes—without proportional cost escalation. Internally, it could enable non-linear revenue expansion of 1.3–1.6x per account, by scaling reusable assets across 50+ GCC relationships, accelerating deal velocity, and improving win rates through differentiated, agent-driven propositions.

In essence, this operating model re-sharpens GCC delivery—decoupling growth from headcount, standardising execution through agentic assets, and embedding continuous optimisation loops—creating a high-efficiency, high-margin, and inherently scalable GCC business model for the future.

*Note: All percentages are derived from PwC analysis based on secondary research.

About PwC

We help you build trust so you can boldly reinvent

At PwC, we help clients build trust and reinvent so they can turn complexity into competitive advantage. We're a tech-forward, people-empowered network with more than 364,000 people in 136 countries and 137 territories. Across audit and assurance, tax and legal, deals and consulting, we help clients build, accelerate, and sustain momentum. Find out more at www.pwc.com.

PwC refers to the PwC network and/or one or more of its member firms, each of which is a separate legal entity. Please see www.pwc.com/structure for further details.

Contact us

Rajesh Ojha
Partner, GCC Leader
rajesh.ojha@pwc.com

Dheeraj Gangrade
Partner and Leader, Consulting-GCC
dheeraj.gangrade@pwc.com

Himadri Ganguly
Partner, Data and Analytics
himadri.ganguly@pwc.com

Rahul Dutta
Managing Director, Advisory
rahul.y.dutta@pwc.com

Contributors

Vishnupriya Sengupta, Sumeet Jalan, Ananya Bandyopadhaya, Ruchika Uniyal and **Siddharth Salar**

Editorial

Dion D'Souza

Design

Kirtika Saxena

pwc.in

KS/April 2026 - M&C 52743