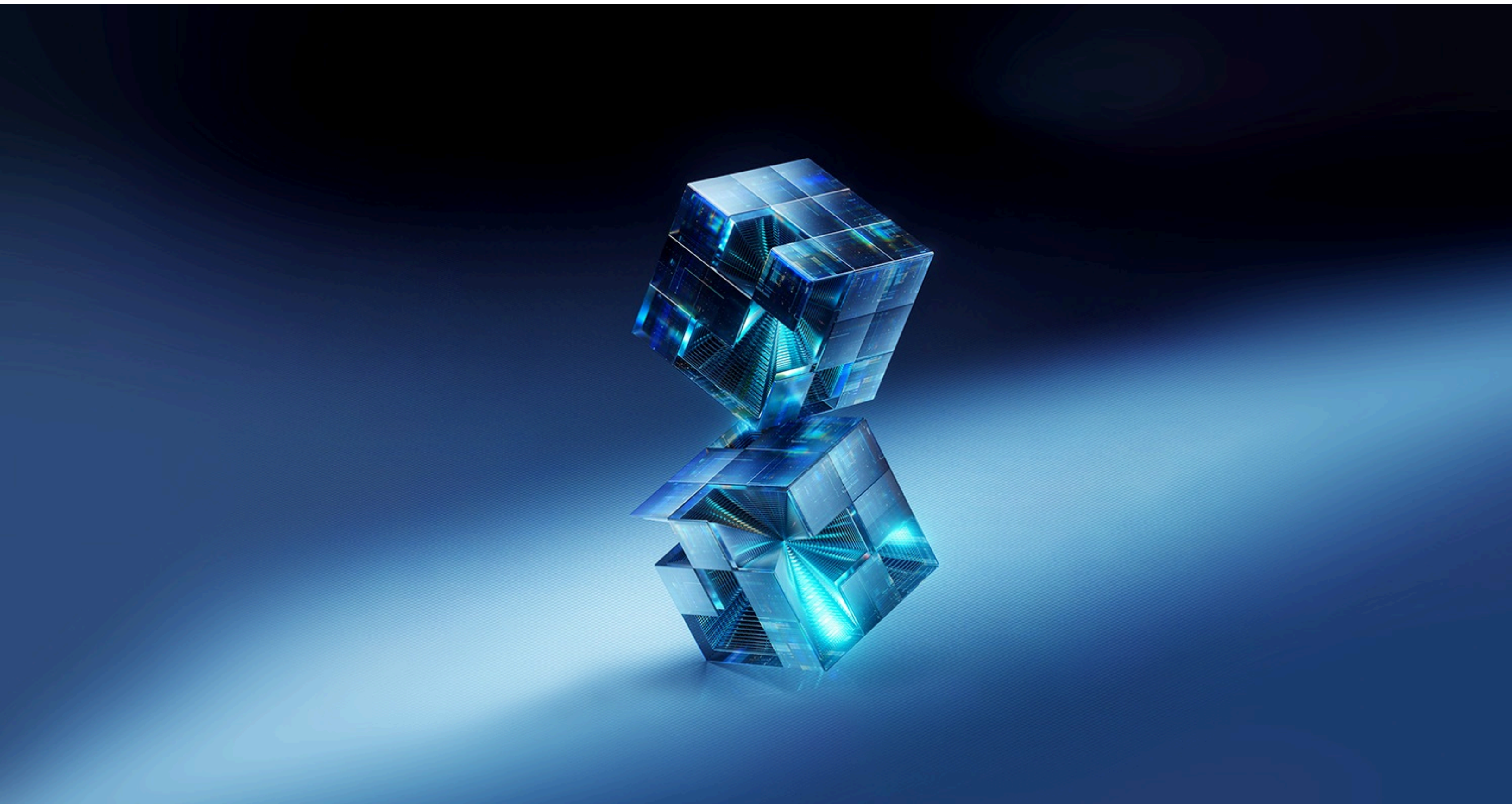
McKinsey
& Company

McKinsey Technology

Overcoming two issues that are sinking gen AI programs

Hard experience has revealed common technology pitfalls in building a gen AI capability and proven strategies for overcoming them.

by Curt Jacobsen, Erik Witte, Kaz Kazmier, and Oscar Villarreal



June 2025

Growth in the generative AI era looks like a classic case of “two steps forward, one step back.” As companies come to grips with the unique complexities of gen AI, initial progress leads to reversals and redos, in some cases threatening to stop development altogether.

There are many sources of frustration and delay, from a lack of sufficient talent to ongoing data quality issues. But our experience working closely with more than 150 companies on their gen AI programs over the course of two years reveals that two hurdles along the building journey almost always surface:

- *Failure to innovate: Process constraints, lack of focus, and cycles of rework that quash innovation.* Teams that could be solving valuable problems are stuck re-creating experiments or waiting on compliance teams, who themselves are struggling to keep up with the pace of development. In our experience, roughly 30 to 50 percent of a team’s “innovation” time with gen AI is spent on making the solution compliant or waiting for their organizations’ compliance requirements to solidify and be practical. Teams work on problems that don’t matter, duplicate work, and create one-off solutions that can’t be reused and often fail to unlock real value.

- *Failure to scale: Risk concerns and cost overruns that choke off scale.* For the few solutions that show real value potential, enterprises largely fail to cross the chasm from prototype to production. Security and risk concerns (including reputational risk) when scaling gen AI applications are handled individually and become too large and expensive to overcome.

Often, these issues happen sequentially as companies try to go from pilot to operational scale, though in some cases, companies can run into these hurdles at the same time. Unfortunately, they can quickly derail entire gen AI programs, not just single applications. While the risk and effort associated with incorrect outcomes or hallucinations are part of the process, premature failures (with examples ranging from messaging inconsistent with branding to violations of policies) can trigger outsize concerns among executives who aren’t suitably prepared or familiar with the testing process. In some cases, these poor test results have led organizations to shut down gen AI programs altogether, which discourages innovation, halts the development of new skills and capabilities, and ultimately leaves the company further away from capturing the intended value it was after.

Whether sequential or simultaneous, these issues are reactions that often boil down to a trade-off between going fast and going carefully. Experience has shown us that this is a false choice. Companies can enable innovation while managing for risk if they are deliberate in building a platform—a centralized set of validated services (for example, ethical prompt analysis, large language model [LLM] observability, libraries of preapproved prompts, multicloud automation, and access controls) and assets (for example, application patterns, reusable code, and training materials) that are easy to find and use (and reuse). Integrating these capabilities into a single platform ensures that products satisfy compliance requirements much more efficiently, which, in our experience, helps to virtually eliminate 30 to 50 percent of the nonessential work typically required.

Our experience building dozens of gen AI solutions for companies across industries has shown that the most successful gen AI platform contains three core components.

1. A self-service portal

Supporting both innovation and scale requires a distributed gen AI capability so that dozens, even hundreds, of teams across the business can easily and securely access tools and services. A secure and compliant self-service portal can meet this necessity in two ways:

— *Developer enablement.* To be effective, this portal and its underlying infrastructure (for example, OpsLevel, Cortex, and Port) should provide a single access point to all validated gen AI products and capabilities. In this way, developers can instantiate preexisting application patterns and begin the development of their specific solution within minutes using approved capabilities that are preconfigured for security and scale. The portal's web interface should incorporate user design principles, such as simple point-and-click processes to provision and deploy gen AI products, as well as provide a well-organized library of documentation and learning modules for all gen AI topics (for instance, how to deploy new resources and how to leverage existing applications) to allow developers to upskill their own capabilities. The best portals allow for contribution models where developers throughout the organization can contribute content and capabilities (such as new libraries and application pattern improvements).

— *Access to management services.* This centralized portal can also provide access to gen AI management services, such as observability and analytics dashboards, as well as built-in budget controls and reporting to prevent cost overruns. Making it simple to follow data access controls, track the governance and approval processes, and understand the current state of applications allows the enterprise to operate hundreds of applications with confidence. These controls can be tailored to environments (for example, lower budgets for ephemeral sandboxes and higher budgets for high-volume testing accounts) and, when integrated with the cost governance components of the AI gateway, can enable teams and IT leaders to monitor ongoing development costs.

2. An open architecture to reuse gen AI services

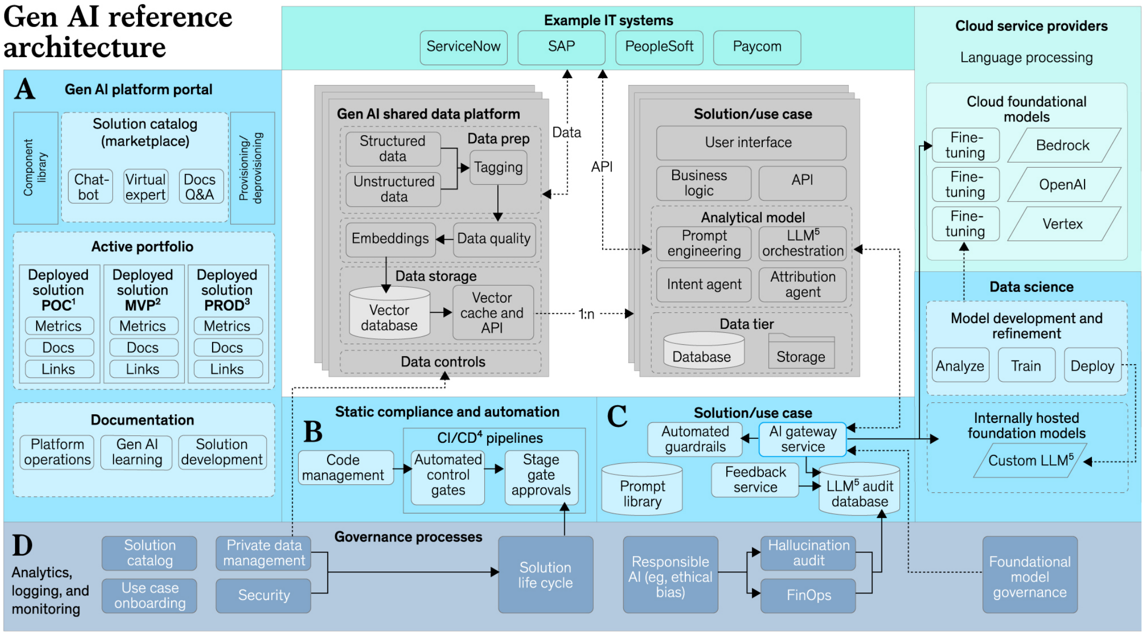
The key to scale in tech is maximizing reuse. Enabling reuse relies on developing an open modular architecture that is able to integrate and easily swap out reusable services and capabilities. This open-architecture approach can also dramatically reduce the total cost of ownership.

Leading enterprises focus on developing two sets of reusable capabilities: the complete gen AI application patterns for common archetypes (such as knowledge management, customer chatbot, or agentic workflows) and data products (for example, RAG and GraphRAG) and the common libraries used in most gen AI applications (for example, chunking and embedding, data reranking, prompt enrichment, or intent classification). Many of these core capabilities can be provided as services.

While it is tempting to turn to a single provider for all gen AI services, that approach often backfires because the provider's capabilities are not suited to all of a company's specific needs and limit access to best-in-class capabilities. With technology rapidly advancing, it makes more sense to use services offered by providers rather than building them (exceptions include building capabilities that directly relate to a company's proprietary advantage). For this reason, the gen AI platform should focus on enabling integration, configuration, and access through an open architecture (exhibit).

Exhibit

Enterprises deploying gen AI at scale follow a common reference architecture.



A **Gen AI platform portal**

Single "pane of glass" for stakeholders built for monitoring, management, and interaction with gen AI applications

B **Gen AI solution automation**

Cloud infrastructure and applications pattern deployments to automate safe and resilient resources that adhere to policies and DevSecOps best practices

C **Gen AI shared services**

AI gateway and associated shared services to provide essential capabilities to all gen AI solutions that help ensure responsible operation

D **Gen AI governance and guardrails**

Robust control framework to provide human users full control of automation and policy adherence (eg, FinOps policies, responsible AI)

[1]Proof of concept. [2]Minimal viable product. [3]Production. [4]Continuous integration/continuous delivery. [5]Large language model.

McKinsey & Company

The core building blocks of an open architecture are infrastructure as code combined with policy as code so that changes can easily be made at the core and adopted quickly and easily by solutions running on the platform. The libraries and component services offered by the platform should be supported by a clear and standardized set of APIs to coordinate calls on gen AI services.

3. Automated, responsible AI guardrails

To mitigate risk, manage ongoing compliance, and provide cost transparency, the gen AI platform should implement automated governance guardrails. One example is having microservices that are automatically triggered during specific points along the software development life cycle or solution operations to review code for responsible AI. These guardrails should automatically audit LLM prompts and responses to prevent data policy violations (such as use of personally identifiable information [PII] in input data), validate compliance of LLM outputs (that is, detect hallucinations, data leakages, and ethical biases), track cost implications (such as of LLM inference and vector database queries), and provide the ability to attribute costs back to the individual solutions. When done correctly, the platform can assist in the automation of data preparation (for example, curation, storage, and risk controls) using data pipelines that allow users to trace a given LLM response back to the original source data.

This approach ensures compliance and more effective cost management and accelerates the secure implementation of use cases by helping application teams spend their time building products and services rather than dealing with the minutia of security and scale. For a large oil and gas company, this approach accelerated the provisioning of new gen AI environments for the development of new applications from over six weeks down to having a dedicated, fully functional gen AI innovation sandbox available in less than a day. Additionally, a gen AI platform accelerates the approval processes by as much as 90 percent because review teams can quickly validate that applications are using approved, shared services.

One of the most effective ways to implement these guardrails is through a centralized AI gateway service that teams and applications are required to use to access LLM models. This AI gateway manages access to approved LLM models based on policies (for example, access controls aligned to user and data classifications combined with the application's environment such as development, test, and production), provides cost attribution (for instance, report of foundational model cost by department), and logs all LLM prompts and responses for follow-on analysis.

The ideal AI gateway uses a flexible system that allows different tools or solutions (that is, middleware) to automatically adjust how they process requests. This includes options to route deployments through various security or policy filters while allowing for exceptions when necessary. For example, one company had a generative AI tool for HR that needed access to sensitive PII to function. In this case, the gateway allowed the PII filter to be temporarily turned off for that specific tool while maintaining protections for other tools.

To manage the unique gen AI problems of hallucinations and ethical biases, the AI gateway keeps a request and response audit log that can be used with parallel models to detect such problems and provide notification to relevant teams or support the ability to automatically fix the model.

Implementing this kind of unique platform requires discipline and time. But the payoff in terms of value and speed over time more than makes up for it, often breaking even after deploying a few solutions. This platform-based approach is essential to accelerating innovation, operating at scale, avoiding common but debilitating technical pitfalls, and allowing companies to capture the promise of gen AI.

Curt Jacobsen is a partner in McKinsey's Southern California office, **Erik Witte** is a partner in the Bay Area office, **Kaz Kazmier** is a partner in the Seattle office, and **Oscar Villarreal** is a partner in the Denver office.

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