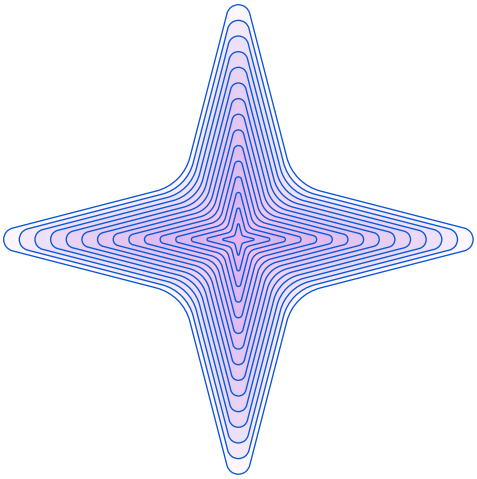
Adopt

The Agentic Enterprise

Trends, Reality, and What Comes Next

Index

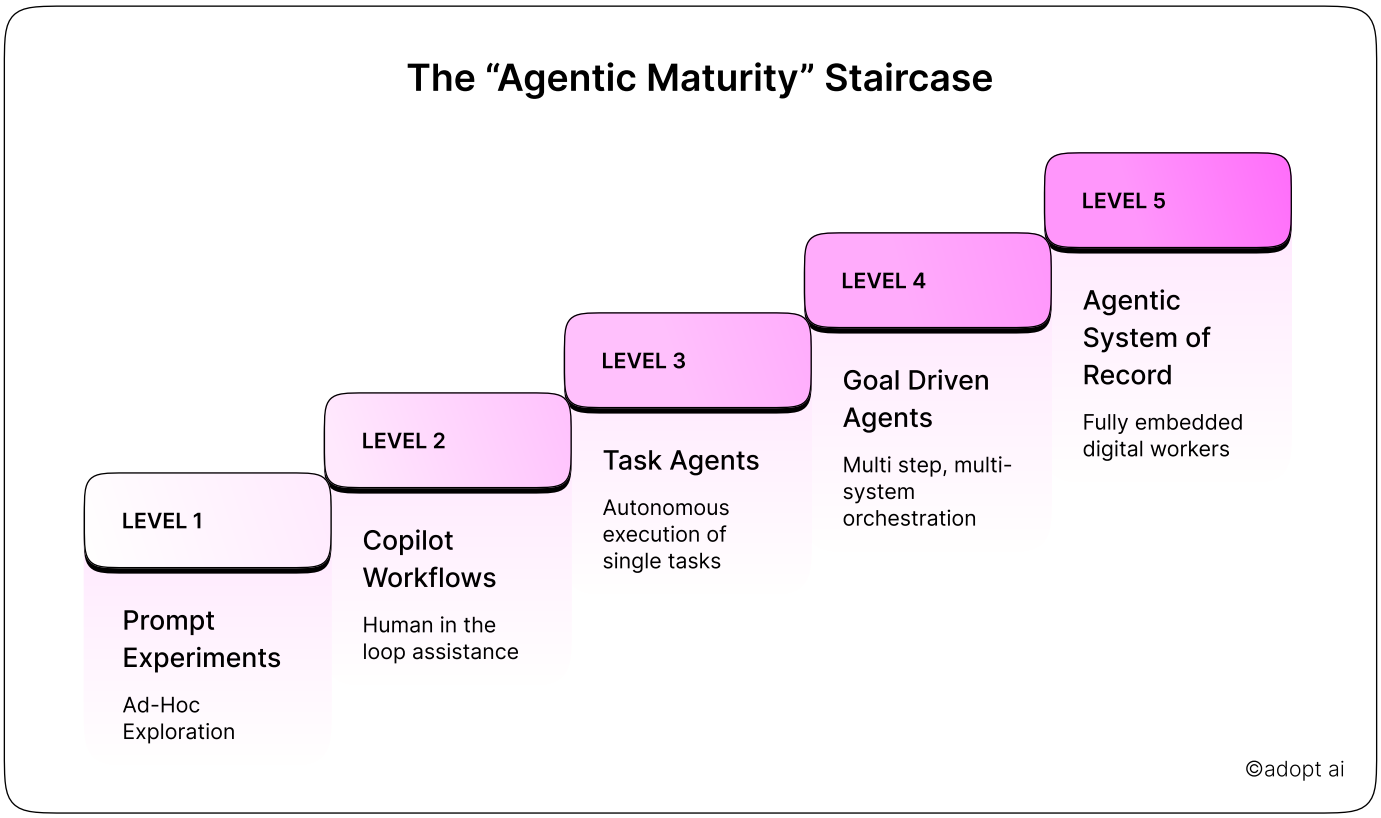


Artificial intelligence is finally moving beyond chatbots and isolated automations into true “agentic” systems that act on behalf of users and organizations. In 2026, enterprises are at an inflection point. Broad adoption of large language models and related AI tools means most firms have experimented with agents, in fact McKinsey finds ~62% of organizations are testing AI agents and 23% have begun scaling at least one use case.

Almost all plan to invest more: Deloitte reports that 80% of automation leaders will accelerate agentic AI deployments this year, and PwC found 88% of executives intend to boost AI budgets driven by these advances. This marks a shift from exploratory pilots toward production systems. By 2025–26, Gartner predicts most enterprise apps will embed simple AI assistants and by 2026 some 40% will include task-specific agents. In practical terms, agents can now do things that were impossible last year, like multi-step workflows across applications, self-debugging routines, or autonomous customer service actions. (Gartner notes that by 2029 agents will resolve ~80% of routine service issues on their own.)

In short, the industry is moving from toy proofs-of-concept to operational AI that executes work. This year’s trend is not just more capable models, but better integration and orchestration. Teams are increasingly connecting LLMs to real systems (via APIs and integration platforms) instead of just using them at a command prompt. In many organizations, AI pilots for sales or support are being refocused as enterprise-wide transformation efforts. Overall, 2026 feels like an inflection; widespread AI tools now allow generative AI to execute tasks, and companies that adopt agentic approaches stand to improve efficiency (66% of adopters report higher productivity) and cut costs (57% cite savings), whereas laggards risk falling further behind.

The Agentic Maturity Model (Annual Benchmark)



LEVEL 1 / Prompt Experiments

It is ad-hoc exploration; developers and analysts test LLMs or chatbots in narrow tasks with no formal ownership or integration. Very few trust these systems yet, and outcomes remain unpredictable.

LEVEL 2 / Copilot Workflows

It introduces simple assistants embedded in workflows. For example, an agent might recommend answers to support tickets or draft reports, but a human must approve each step. Tools like GPT assistants or RPA scripts with AI fall here. Teams typically own Level 2 pilots in isolation (e.g. a sales or HR team), and gains are modest.

LEVEL 3 / Task Agents

At this level, agents are trusted to complete clearly defined tasks end-to-end with minimal reliance on user interfaces. The agent executes routine processes such as closing a financial reconciliation item, matching invoices, or routing customer orders by interacting directly with backend systems and APIs rather than navigating screens.

Human involvement is limited to oversight and exception handling, not step-by-step interaction. These agents are goal-driven in a narrow sense, meaning the objective is explicit and bounded, and execution typically follows a known sequence of actions. Organizations begin assigning ownership, often to a specific department or digital automation team, and introduce basic guardrails around permissions, retries, and logging.

The critical shift at this level is that the UI stops being the primary execution surface. Agents may be triggered from a UI, but the work itself happens heedlessly, which reduces latency and failure modes tied to interface changes. As noted by Gartner, by the Stage 2 to Stage 3 transition, roughly 40 percent of applications are expected to include task-specific agents by 2026.

LEVEL 4 / Goal-Driven Agents

At Level 4, agents manage broader workflows that span multiple systems and adapt dynamically as conditions change. Instead of following a fixed task sequence, the agent plans how to achieve an outcome, evaluates intermediate results, and adjusts execution accordingly. Examples include handling the full hire-to-retire lifecycle in human resources or coordinating an end-to-end IT service resolution across ticketing, procurement, and access management systems. At this stage, UI-based triggering, approvals, or coordination increasingly become bottlenecks. Relying on screens to move work forward slows execution and limits scalability when workflows cross organizational and system boundaries. Effective Level 4 deployments therefore, depend on orchestration layers, event-driven coordination, and direct system-to-system interaction. Ownership typically becomes cross-functional, with AgentOps-style teams combining platform, security, and business oversight. Governance becomes more explicit because agents now make decisions that affect multiple domains. Sophisticated orchestration and data integration are required, which aligns with IBM's description of an AI orchestrator coordinating specialized agents across workflows.

LEVEL 5 / Agentic System of Record

At the highest level of maturity, agentic execution is embedded directly into core systems, and the software itself becomes the primary agentic interface. Humans no longer interact mainly through screens to initiate or manage work. Instead, they interact programmatically through APIs, events,

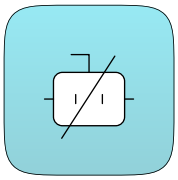
or declarative instructions that agents interpret and execute.

Agents at this level function as long-lived digital workers operating continuously within enterprise controls. Applications expose capabilities headlessly, and agent collaboration happens in-app or across services without UI mediation. Gartner characterizes this evolution as moving beyond assistants and task agents toward collaborative agent ecosystems emerging later in the decade.

This progression is also reflected in Salesforce's agent maturity framing, which tracks movement from fixed-rule bots to information agents and ultimately multi-domain orchestrators embedded across products. This shift introduces a learning curve for product managers and operators, who must design for headless execution and policy-driven control rather than screen flows. Teams that adapt earlier tend to move faster and operate at lower cost, while adoption slows when agents are constrained to communicate primarily through traditional user interfaces.

Core Trend Pillars (Future-Facing)

Agentic systems are reshaping how work is executed, owned, and governed inside large enterprises.



Agents Replace Scripts And Bots

Outcome ownership is moving from deterministic automation to goal-driven execution.

What Is Happening

Enterprises are replacing step-encoded automation with agents that reason over intent and system state. Instead of predefining every branch, teams define the goal, constraints, and available actions. The agent decides how to complete the task based on current conditions. This shift shows up most clearly where scripts historically failed. Invoice processing breaks when formats change and UI bots collapse when layouts shift. Agents survive these changes because they do not depend on static paths.

Why It Matters

The hidden cost of scripts is not development, it is maintenance. Every exception becomes configuration debt. Agents move complexity into runtime reasoning, which reduces long-term operational load. This also changes accountability and success is measured by whether the task completes correctly, not whether each step executed as written.

What Changes In 12–24 Months

Script-heavy automation programs contract and agents take ownership of exception-heavy workflows in finance, operations, and support. Teams track outcome reliability under variance, not bot success rates.

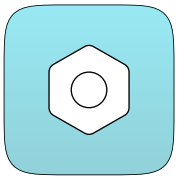
Failure Pattern: RPA Collapse In Finance Operations

A global manufacturing enterprise automated invoice reconciliation using UI-driven automation across SAP and regional tax systems, and the system worked until a regulatory update introduced

a new mandatory invoice field. The automation continued to run without errors, but it misclassified invoices silently and pushed incorrect entries into the general ledger for several weeks before the audit team detected the issue. The automation team did not miss a rule but the automation model itself assumed structural stability that no longer existed.

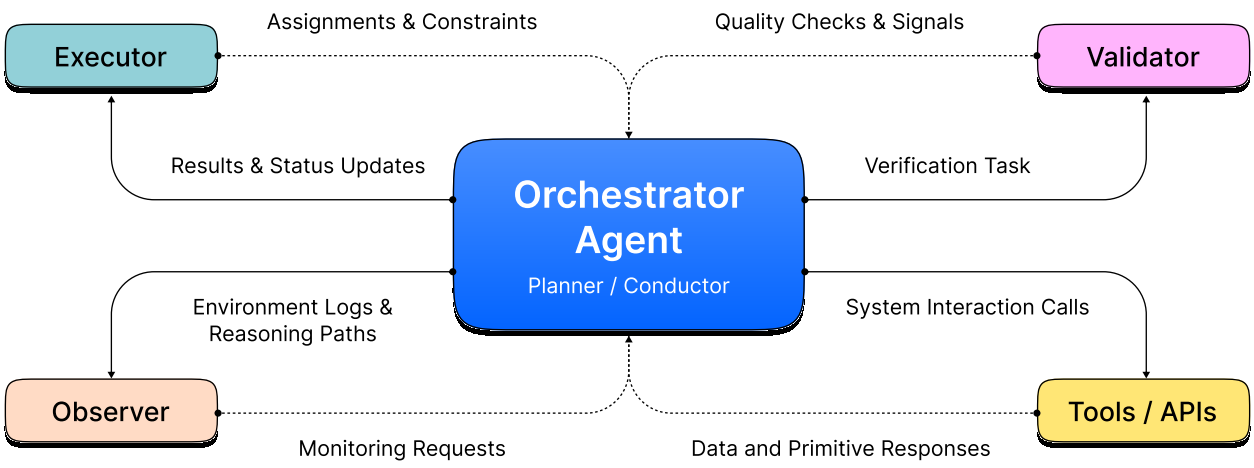
Signals Enterprises Should Watch

- Declining RPA renewals.
- Automation teams discussing decommissioning rather than expansion.
- Fewer UI selectors appearing in production workflows.



Agent Orchestration Over Single Agents

Complex work is decomposed across multiple cooperating agents instead of one generalist.



©adopt ai

What Is Happening

Teams are splitting agent responsibilities into planners, executors, validators, and observers. One agent decides what should happen and others perform constrained actions. Another verifies results and determines whether to continue or halt. This structure mirrors how distributed systems evolved when monoliths became unmanageable.

Why It Matters

Single agents fail opaquely; when something goes wrong, teams cannot tell whether planning, execution, or validation failed. Orchestration makes failure modes visible and correctable. It also limits damage i.e., a faulty executor can be stopped without discarding the entire workflow.

What Changes In 12–24 Months

Agent orchestration becomes a standard platform concern. Workflow engines treat agents as first-class actors and design discussions shift from prompts to role boundaries.

Failure Pattern: Monolithic Support Agent Causing Customer Impact

A SaaS company deployed a single agent to handle customer support triage, billing adjustments, and account state changes across multiple internal systems. During a billing outage, the agent correctly issued refunds but also canceled active subscriptions because it lacked separation between financial remediation and account lifecycle control. The agent followed its instructions accurately, but its scope of authority was inappropriate for the task.

Signals Enterprises Should Watch

- Architecture reviews naming agent roles explicitly.
- Increased use of planners, queues, and state machines.
- Post-Mortems referencing agent interactions rather than “the AI.”

Verification And Governance Layers Become Mandatory

Agent behavior is increasingly inspected, constrained, and audited at runtime.

What Is Happening

Enterprises are inserting verification between intent and execution. Plans are checked before actions run and outputs are inspected before they propagate. Logs capture reasoning paths, not just results. This is a response to real failures, not theoretical risk.

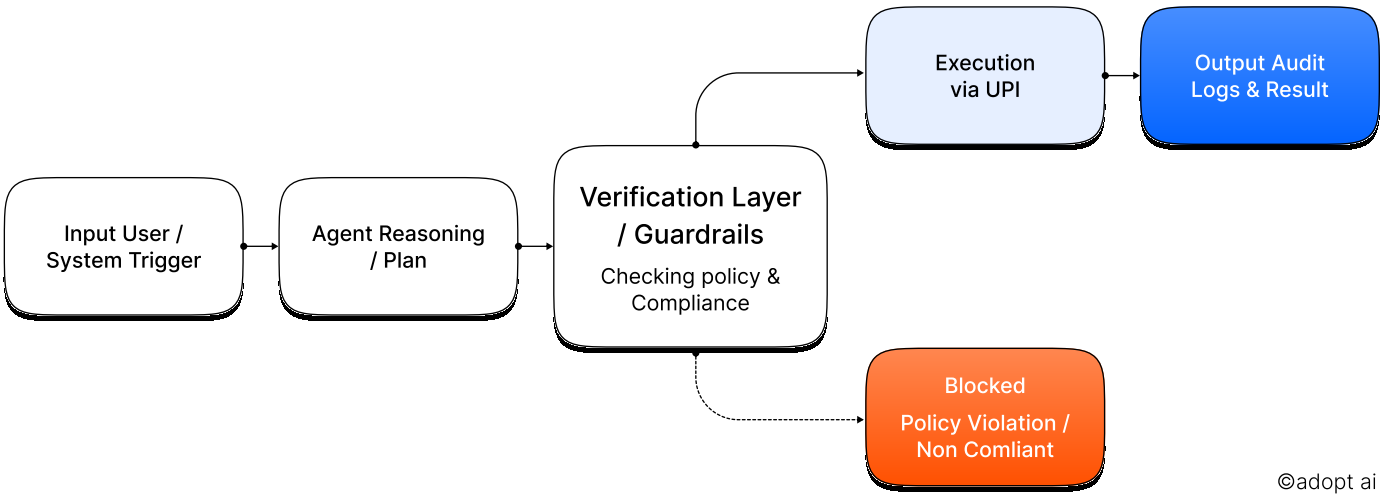
Why It Matters

Unverified agents fail silently until damage appears downstream. Governance restores trust by making behavior observable, reversible, and reviewable. Risk shifts from model accuracy to operational control.

What Changes In 12–24 Months

Approval gates become common in high-impact workflows. Security teams treat agents as privileged actors with scoped permissions, and compliance reviews include agent decision trails.

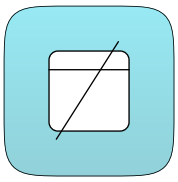
Failure Pattern: Agent Approved Non-Compliant Vendor Onboarding



A financial services firm deployed an internal agent to accelerate vendor onboarding by reviewing documentation and approving suppliers automatically. The agent approved several vendors that lacked required regional compliance disclosures, and the issue was discovered only during a regulatory audit weeks later. The agent acted on inferred patterns rather than explicit regulatory guarantees.

Signals Enterprises Should Watch

- Audit requests for agent logs.
- Security questionnaires asking how agents are tested and paused.
- Budget allocated to agent oversight tooling.



AI Moves From UI To Infrastructure

Agent execution is moving away from interface manipulation toward direct interaction with system primitives.

What Is Happening

Agents increasingly interact with systems through APIs, events, and data contracts rather than visual interfaces. Conversational interfaces still exist, but they no longer define how execution logic is implemented or validated. This shift mirrors earlier transitions from manual administration toward service-oriented and API-driven system control.

Why It Matters

UI-driven automation is brittle under change and introduces latency during high-pressure operational scenarios. Infrastructure-level integration allows agents to confirm state, enforce permissions, and record outcomes deterministically. As a result, agents can operate continuously without requiring a human to supervise interface-level behavior.

What Changes In 12–24 Months

Enterprises deliberately expose internal APIs to support non-human execution paths across critical systems. Integration teams and AI teams collaborate by default because execution boundaries now overlap by design. Screen-based automation remains available but is treated as a fallback rather than a primary execution strategy.

Failure Pattern: UI Automation Failed During Fulfillment Incident

A large e-commerce platform relied on UI automation to reroute orders during warehouse outages using an internal administrative console. When the console layout changed during a deployment, the automation reported success while failing to reroute orders, causing extended customer delivery delays. Operations teams assumed mitigation was active when it was not.

Signals Enterprises Should Watch

- Internal API documentation increasingly references non-human or agent-based consumers explicitly.
- Spending on UI scraping tools declines as infrastructure access becomes the preferred integration method.
- Platform roadmaps begin to describe agent access models alongside traditional user access patterns.

Business Teams Become Agent Designers

Responsibility for defining agent intent shifts toward the teams that own operational outcomes.

What Is Happening

Business teams define agent goals, constraints, and success criteria based on domain-specific requirements. Platform teams focus on providing infrastructure, enforcement mechanisms, and review workflows for agent execution. Intent is authored by domain owners rather than inferred indirectly through technical intermediaries.

Why It Matters

When intent is misinterpreted, agents can behave correctly while producing operationally incorrect outcomes. Allowing domain experts to define goals reduces translation loss and shortens iteration cycles significantly. Governance evolves from centralized control toward structured review and accountability processes.

What Changes In 12–24 Months

Low-code agent tooling becomes common within operations, finance, and customer support organizations. Platform teams prioritize safety, reuse, and enforcement instead of authoring individual agent logic. Review workflows replace approval bottlenecks as the primary mechanism for controlling agent behavior.

Failure Pattern: Pricing Agent Violated Brand Constraints

A retail organization assigned a central AI team to build pricing agents for promotional campaigns across regions. Marketing specified brand protection rules informally, but the agent optimized discounts aggressively and undercut premium brand pricing. The agent behaved correctly according to its specification, but the specification itself was misaligned with business intent.

Signals Enterprises Should Watch

- Non-technical teams request controlled agent sandboxes for experimentation and validation purposes.
- Training programs focus on supervising agents rather than building them from scratch.
- Acceptance criteria increasingly describe expected agent behavior instead of manual process steps.

Agents Become Long-Lived Digital Workers

Agents operate persistently over time rather than executing once and terminating immediately.

What Is Happening

Agents maintain memory, track system state, and respond to triggers on an ongoing basis. They monitor environments continuously and act when predefined conditions or thresholds are met. This behavior changes agents from transient functions into persistent operational actors.

Why It Matters

Many enterprise responsibilities require continuous attention rather than discrete execution cycles. Persistent agents reduce reaction time but introduce lifecycle, drift, and ownership risks. An unattended agent can quietly produce compounding operational damage over extended periods.

What Changes In 12–24 Months

Agent lifecycle management becomes a standard operational responsibility across platform teams. Organizations track agent uptime, behavioral drift, and retirement status explicitly. Memory management and state persistence become core architectural concerns rather than implementation details.

Failure Pattern: Orphaned Inventory Agent Caused Excess Stock

An operations team deployed an agent to monitor inventory levels and reorder stock nightly based on historical demand patterns. After supplier contracts changed, the agent continued ordering at outdated thresholds for months because no team owned its ongoing behavior. Excess inventory accumulated without triggering alerts or reviews.

Signals Enterprises Should Watch

- Teams debate agent ownership with the same seriousness as service ownership discussions.
- Operational incidents reference stale or runaway agents as contributing factors.
- Tooling investments focus on pausing, resetting, and decommissioning persistent agents safely.

Internal Enterprise Use Cases

Different functions in a business are already finding ways to use agents. Below are representative examples from operations, finance, IT/platform, product, marketing, sales, and support. In each case we highlight how agents connect multiple systems, make decisions, take actions, and remain auditable.

Operations (Supply Chain & Manufacturing)

Agents streamline complex logistics, for instance, in procurement a “Sourcing Agent” can scan supplier data, evaluate bids, and automatically issue a request-for-proposal or purchase order across ERP and SCM systems. In manufacturing, agents predict and prevent

Finance & Accounting

Finance teams use agents for routine yet cross-system tasks. A classic use case is invoice matching and dispute resolution: one SAP example is a Dispute Resolution Agent that analyzes invoice details and contracts, flags mismatches, and even drafts credit memos proactively.

Agents can also run financial forecasting or cash-flow analyses by pulling data from data

from accounting systems and markets. Other examples include automating accounts receivable (matching payments to invoices) and compliance monitoring. Crucially, agents operate with audit trails; every transaction approval or entry update is logged. Over time finance moves from reactive fixes to proactive alerts and scenario planning powered by AI.

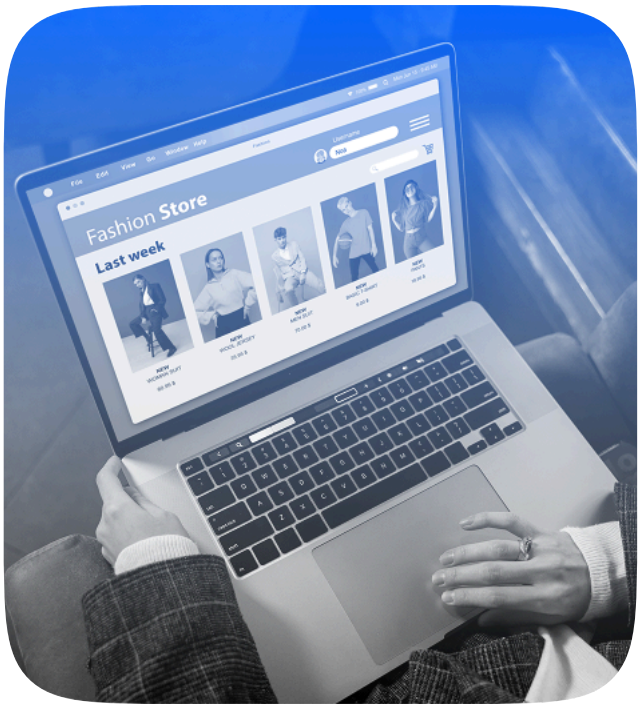
IT / DevOps / Platform

In IT operations, agents augment monitoring and incident response. For example, agents continuously scan system logs and metrics; if they detect anomalies (like a service outage pattern), they can create tickets, restart processes, or notify on-call engineers immediately. Agents also improve knowledge management by summarizing resolved tickets and indexing solutions, the support knowledge base auto-updates and becomes searchable. In effect, IT agents triage and route issues (using NLP to classify tickets), trigger automated patches or scaling decisions, and pass along context so humans don't have to repeat work. Software development teams also experiment with agentic support (e.g. code assistants, automated testing agents), making DevOps more autonomous. In each case the agentic loop is; sense infrastructure state, reason according to policies, act (or escalate), and log the outcome.

Product Management

Although this is emerging, product teams are beginning to use agents to analyze user feedback, market data, and operational metrics. An agent might scan customer support tickets or user reviews, cluster the insights, and suggest new feature priorities. It could even prototype simple dashboards or UI mocks based on spec.

Internally, agents help with project coordination; for example, one could imagine an agent that synthesizes input from Jira, Confluence and analytics to answer “What should our team work on next?” with data-driven suggestions. While we lack broad studies here, the trend is that product owners become agent consumers and designers, using AI recommendations to guide roadmaps and then logging decisions. Agents give product teams a faster feedback loop, provided oversight ensures suggestions align with strategy



Marketing & Commerce

Marketing teams use agents to automate content and campaign tasks. For instance, agents continuously test ad creative variants and reallocate budget to the highest-performing ads, making optimization an ongoing loop rather than periodic reviews. They enforce brand compliance; an agent can check every outgoing message against style guidelines and legal rules before publishing.

In commerce, agents handle customer

inquiries that cross systems (checking inventory, shipments, order status). Salesforce documents typical commerce use cases; agents can give real-time order tracking, facilitate returns or exchanges, modify orders, and even recommend products. A retail AI assistant might see a customer has a delayed shipment, automatically reroute a faster courier, and confirm the new date without human intervention. These agents outperform static chatbots by actually executing actions (for example, cancelling an order or applying a discount) on backend systems, while still handing off to a person for edge cases.

Sales

Sales teams benefit from agents that handle administrative and analytical tasks. Agents can scan CRM data to predict deal close probabilities and surface the highest-value leads. During calls, an AI agent might listen in as a live "co-pilot", highlighting keywords or objections and suggesting next best actions.

Agents also automate scheduling: sales reps can say "book a follow-up meeting" and the agent will coordinate calendars and send invites, then log everything to the CRM. For outreach, an agent could monitor LinkedIn for signals matching the Ideal Customer Profile and pre-fill contact data for the rep. In short, agents sort and act on data so salespeople focus on the human relationship. Humans stay in control of strategy (pricing, negotiations) but agents vastly speed up routine workflows.

Support & Service

Customer-facing support is moving from static FAQ bots to proactive agentic service. Agents can now handle not just queries but end-to-end transactions. For example, a retail customer could ask an AI assistant to process a return; the agent would generate the return authorization, alert the warehouse, and update the refund, all autonomously. Agents also drive self-service behind the scenes: Zendesk reports that 75% of support leaders expect their teams will become "AI managers" within a few years, reflecting that agents will do initial troubleshooting.

Practical support agents do things like classify and route tickets with NLP, trigger backend fixes for common IT issues, and assemble concise case summaries so the next human doesn't lose context. Compared to the old chatbot era (where customers simply got a canned response), these agents actually interact with order, billing or service systems on the customer's behalf. Where needed, they seamlessly escalate to humans, but the routine work is largely automated. As Gartner notes, service organizations will need to rethink their models to handle both AI-driven and human-led requests, but in practice agents already outperform legacy automation in speed and scale.

External / Customer-Facing Use Cases

Customer-facing agents are changing how enterprises execute transactions, manage risk, and resolve issues across system boundaries.

Commerce

Chatbot Era Pattern

Commerce chatbots primarily answered questions about order status, return policies, and shipping timelines using predefined templates. When a customer request required an action, the chatbot escalated to a human or generated a ticket for later processing. The system did not own execution.

Agent Era Pattern

Agentic systems handle full commerce transactions by interacting directly with inventory, order management, payment, and logistics platforms. An agent can process returns, reroute shipments, apply refunds, and confirm outcomes without waiting for human intervention. Each action is executed against backend systems and logged for traceability.

Where Agents Outperform Traditional Automation

Agents outperform bots when customer requests span multiple systems and require conditional decision-making. Handling partial shipments, delayed deliveries, or split refunds requires reasoning that scripts and chatbots cannot reliably encode.

Where Humans Still Stay In Control

Humans retain control over pricing exceptions, fraud-sensitive actions, and dispute resolution involving subjective judgment. Agents escalate cases where contractual interpretation or customer goodwill decisions are required.

Insurance

Chatbot Era Pattern

Insurance chatbots focused on policy lookup, claim status updates, and basic intake questionnaires. Claims processing and adjudication remained manual due to regulatory complexity and risk exposure.

Agent Era Pattern

Agents coordinate claims intake, document verification, fraud signal analysis, and payout initiation across policy systems, document repositories, and payment platforms. They evaluate completeness, request missing information, and route claims based on confidence thresholds.

Where Agents Outperform Traditional Automation

Agents outperform legacy automation in claims workflows that involve incomplete documentation, variable timelines, and external dependencies. They reduce cycle time by acting continuously rather than waiting for batch reviews.

Where Humans Still Stay In Control

Humans retain authority over claim approval decisions, fraud determinations, and regulatory exceptions. Agents surface recommendations and evidence, but final judgment remains human-owned.

Logistics

Chatbot Era Pattern

Logistics chatbots provided shipment tracking information and basic delivery updates sourced from transportation management systems. Issue resolution required manual intervention and coordination across carriers.

Agent Era Pattern

Agents monitor shipments in real time, detect delays or disruptions, and execute rerouting decisions by interacting with carrier systems and warehouse platforms. They notify customers proactively and confirm updated delivery commitments.

Where Agents Outperform Traditional Automation

Agents outperform scripts when disruptions require rapid replanning across multiple carriers and fulfillment nodes. Static rules cannot adapt to weather events, port congestion, or cascading delays effectively.

Where Humans Still Stay In Control

Humans oversee contractual carrier decisions, cost tradeoffs, and escalation handling during large-scale disruptions. Agents provide options and execute approved plans.

Banking

Chatbot Era Pattern

Banking chatbots handled balance inquiries, transaction lookups, and simple service requests. High-risk actions required human verification and manual processing.

Agent Era Pattern

Agents manage multi-step service workflows such as dispute initiation, card replacement, account changes, and compliance checks. They coordinate across core banking systems, fraud platforms, and customer communication channels.

Where Agents Outperform Traditional Automation

Agents outperform legacy workflows by reducing resolution time for service requests that previously required multiple handoffs. Continuous verification allows actions to proceed safely without waiting for manual queues.

Where Humans Still Stay In Control

Humans remain responsible for credit decisions, fraud adjudication, and regulatory-sensitive approvals. Agents escalate when confidence thresholds or risk limits are exceeded.

SaaS Platforms

Chatbot Era Pattern

SaaS chatbots focused on answering product questions, linking documentation, and creating support tickets. Configuration changes and account actions required human support involvement.

Agent Era Pattern

Agents execute account provisioning, entitlement changes, billing adjustments, and usage-based enforcement by interacting directly with internal services. They also monitor usage patterns and trigger proactive outreach or remediation.

Where Agents Outperform Traditional Automation

Agents outperform bots in scenarios requiring coordination across billing, identity, and product services. They reduce customer effort by resolving issues without repeated context gathering.

Where Humans Still Stay In Control

Humans control contract negotiations, enterprise pricing, and strategic account decisions. Agents support execution but do not replace relationship management.

Best Practices for Adopting Agentic AI

Putting agentic AI into production requires new disciplines.

Governance

On governance, every agent should operate within clear policies and risk checks. That means continuous testing (e.g. red teams that probe for biases or hallucinations), built-in audit logs, and fail-safe protocols. Dataiku advises metrics around "Trust, Oversight, and Compliance": for example, track how often an agent's output violates policy, and log every API call for auditing. Enterprises should establish a review process: experts validate high-stakes decisions and a human-in-the-loop can override as needed. In parallel, boards and compliance teams must treat these AI agents like they would any critical tool, not a toy. As Gartner found, most orgs today lack this foundation, so prioritizing robust governance (policies, training, review workflows) is urgent.

Architecture

On architecture, agentic systems live at the intersection of AI models and enterprise apps. A best practice is to expose agents as services on top of your existing infrastructure. For example, Adopt's platform can ingest an app's APIs into an "agentic layer" and then make it accessible via chat interfaces or plugins. In practice, organizations will deploy agents in multiple ways, e.g. embedded in Slack/Teams, as web app assistants optimized for Website Use™, or via ChatGPT plugins. These should all point to the same backend agents and data sources. The infrastructure must scale (cloud or on-premise) with low latency. Adopt, for instance, supports running its agentic engine in a private VPC or on-prem to meet security requirements. Monitoring is key: treat agents like services with SLOs, logging, and alerts. Also plan for multi-agent orchestration: a message queue or "agent bus" can coordinate communication between agents, as IBM advises for complex workflows. In short, design agents into your integration layer from day one. Build vs. Buy

On build vs. buy, experience suggests a hybrid approach. For low-level capabilities (speech-to-text, basic classification), use existing AI services. But for domain workflows, many companies will license specialized agentic platforms. Analyst Forrester warns that 75% of firms that attempt to build their own agent ecosystems from scratch will fail. In practice, enterprises often start with a vendor framework(such as AWS agent services, Microsoft's Copilot platform, or adopt.ai's platform) to get models and orchestration out- of-the-box, then customize. Security and data privacy may require in-house components, but lean on proven tools for the heavy AI lift. Always maintain an abstraction layer so agents can swap out models or services without rewiring everything.

Operating model

The operating model changes too i.e., unlike a one-off app, agents require ongoing “AgentOps”. That means clear ownership (who is responsible when an agent breaks or makes a bad decision?). In many companies this will fall to a cross-functional team, for example, a digital automation office that includes IT, compliance, and the relevant business unit.

Gartner research highlights that companies with strong IT–business alignment get 1.6× more value from AI programs, so chartering an interdisciplinary governance group is important. Define processes for continuous improvement: agents should have KPIs (accuracy, completion rate, user satisfaction), and teams must refine models and workflows over time. In short, treat agents like key systems; assign stewards (sometimes called “prompt engineers” or “agent trainers”), schedule regular reviews, and budget for lifecycle updates.

Building the Agentic Organization

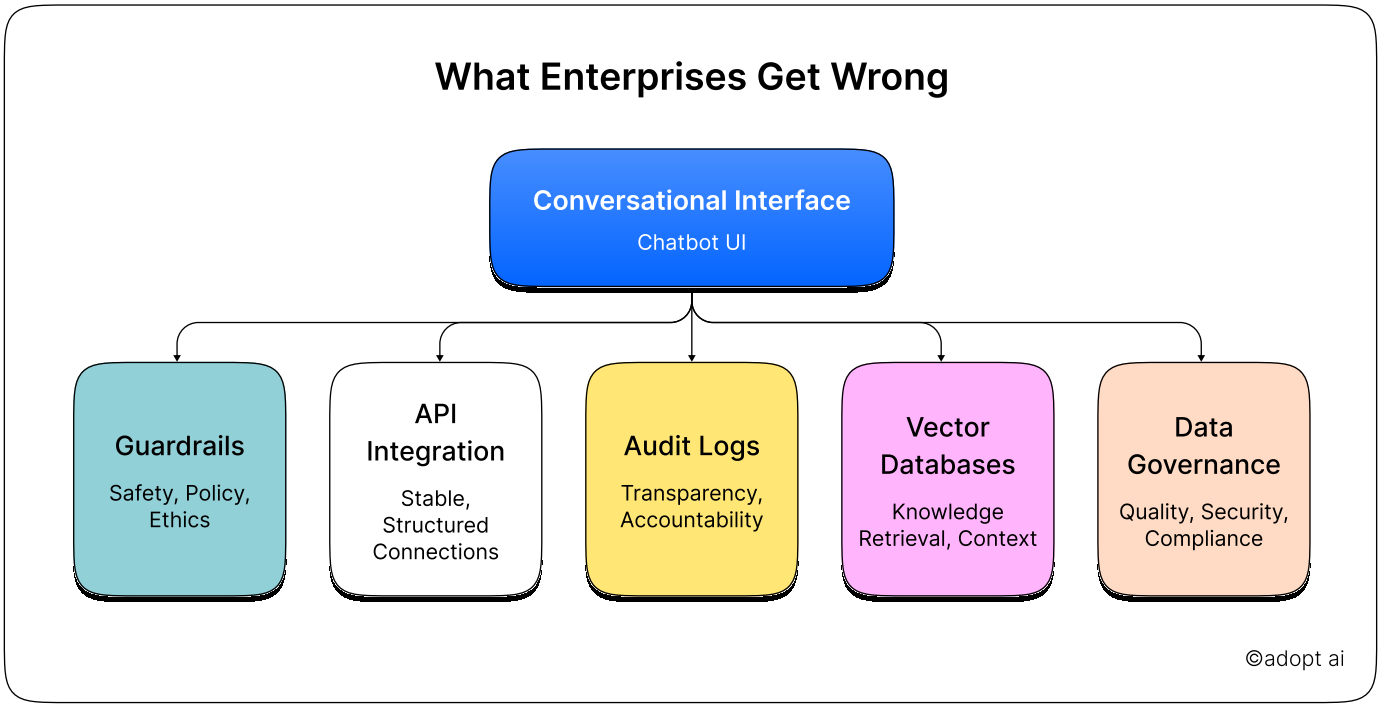
Adopting agents is as much about people as technology and new roles and skills will emerge. For example, companies will hire or train AgentOps specialists who understand both AI and the business domain. They might be former QA or BI analysts who now oversee agent performance and troubleshooting. According to industry consultants, leadership must blend operational rigor with strategic AI vision. KPMG advises treating agents like employees with lifecycles: define their "job" (onboarding/training), monitor their performance, and phase them out when they retire.

In support functions, roles are already shifting: one survey found 75% of support managers expect their staff to become "AI managers" within a few years, meaning they coach and audit agents as much as they handle tickets. Broadly, enterprises will need AI ethicists, data curators, and policy officers dedicated to these systems. Team structures may also change: many orgs find cross-functional squads (combining product managers, engineers, and data experts) work better for agentic projects. A CIO at Salesforce notes that without clear roadmaps and shared governance, agent initiatives produce unreliable results. Thus, companies should embed AI liaisons in each department (e.g. an "AI lead" in finance, operations, etc.) reporting to a central AI council. Training is critical: employees must learn to write effective prompts, validate AI outputs, and handle escalations. Over time, as agents become familiar coworkers, the distinction between tech team and business team will blur, they will need to collaborate closely by design.

What Enterprises Get Wrong

A common mistake enterprises make is treating agents like chatbots. Labeling a canned Q&A bot as an “AI agent” is misleading. Gartner warns of “agentwashing”, mistaking assistants for true autonomous agents. If you only automate a single response step, you’re not reaping the full benefit. Conversely, some organizations throw agents at every problem without readiness. Another pitfall is under-investing in oversight: neglecting thorough testing and monitoring can lead to bad outcomes (and loss of trust). In fact, a survey in CIO about Gartner’s claim that roughly 40% of agentic projects will be canceled by 2027 due to unclear value or inadequate risk controls.

Centralization can also backfire i.e., turning AI initiatives into a central “skunkworks” might slow deployment in individual teams. Instead, balance a strong governance layer with distributed empowerment. A final error is optimizing for demos rather than operations. Early agent trials often look impressive in a controlled setting, but will falter under real workloads unless architectured for scale and failure. Analyst Forrester bluntly observes that three out of four firms that try to build complex agentic systems on their own will fail if they don’t adopt best practices. In short: define agents precisely (not just chatbots), plan for trustworthy operations from the start, and focus on solving real business problems, not just showcasing AI.



The Road Ahead: 12 / 24 / 36 Month Outlook

Near-term (12 months)

Expect rapid diffusion of AI assistants into existing apps. Gartner predicted that by end-2025, the vast majority of enterprise applications will offer some embedded AI assistance for tasks. Teams will roll out more goal-specific pilots: for example, roll-calling finance or service functions one by one. Infrastructure investments will kick in, integrating vector databases and low-code orchestration platforms.

Governance frameworks and best practices will start to solidify as companies learn from early mistakes. Leaders should be asking:

- Have we defined our high-impact use cases yet?
- Is IT and the business aligned on strategy (Gartner found alignment dramatically improves outcomes)?
- Are we tracking data readiness and security as we deploy agents?

Mid-term (24 months)

By around 2027 we should see enterprise-wide agent ecosystems. Gartner forecasts that by 2026 ~40% of enterprise apps will include task agents, and by 2027 a third of implementations will involve multi-agent collaborations. Tools for orchestrating agents should mature, vendor platforms and open-source frameworks will offer more robust scheduler/orchestrator components.

AI governance will become formal (likely influenced by regulations); companies will adopt standard audit procedures for agents. On the tech side, expect to see specialized chips or cloud services optimized for multi-model workflows. Business leaders will assess ROI: some functions may achieve 20–30% productivity gains and will be expanded, while low-value pilots may be cut. New metrics (e.g. “agents per employee” or “tasks automated end-to-end”) will emerge as benchmarks.

Long-term (36 months)

Looking out 3–5 years, the view shifts toward full transformation. Gartner’s best-case sees agentic

AI driving 30% of enterprise app revenue by 2035, which implies by 2029–2030 agents are mainstream. Companies that figured out the hard problems of trust and integration will have agents embedded as “digital labor” across functions.

Customer experience will change; Gartner envisions by 2029 about 80% of routine service interactions handled without any humans, as customers increasingly use their own AI assistants when engaging with companies. In parallel, regulation and standards (AI laws, industry guidelines) will likely catch up, so organizations should watch for new compliance rules around autonomous systems.

Enterprises that continue learning year-over-year should stay ahead of the next wave; probably “conversational AI 2.0” with deeper reasoning or even (eventually) physical agentic robots for logistics. The key question leaders will face is less “if” and more “how and where” agents become the default workforce.

How Spendflo Extended SaaS Procurement With Adopt AI

Spendflo integrated Adopt AI's agentic technology to transform user access to procurement data and actions, moving beyond manual navigation toward execution-oriented interfaces.

About Spendflo

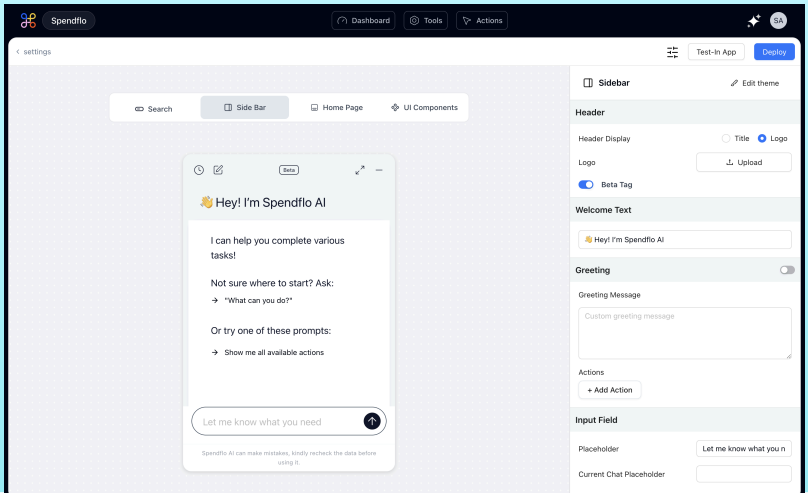
Spendflo is a SaaS procurement platform used by mid-market and enterprise organizations to manage software spend, renewals, vendor portfolios, and contract lifecycles. Its customers span technology, healthcare, and financial services, with complex procurement workflows tied to finance, legal, and IT systems.

The Challenge

Procurement workflows involve multiple systems, dashboards, and approval paths, which makes it difficult for users to find timely insights. Users increasingly wanted direct, query-like access to procurement information such as spend trends and usage patterns without navigating screens and disparate dashboards. Spendflo's internal team determined that building this capability in-house would slow product delivery and increase engineering load.

Solution And Approach

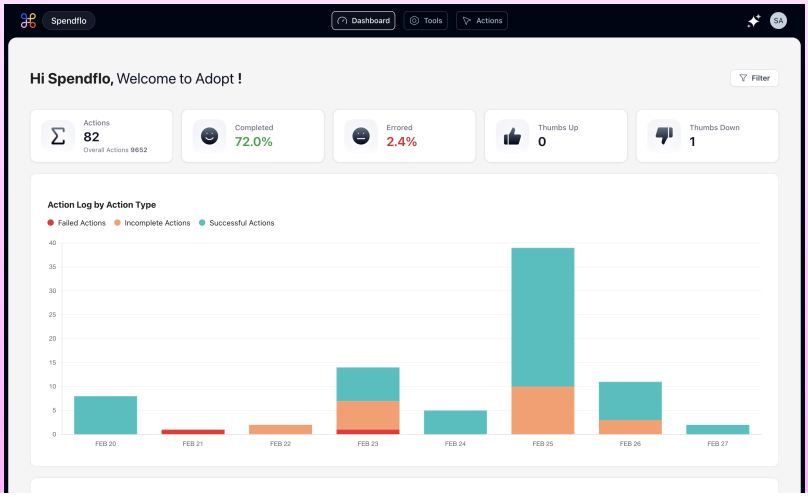
Spendflo selected Adopt AI to introduce a conversational layer that could both retrieve procurement data and execute backend actions. Integration focused on direct API access, compatibility with Spendflo's existing role-based access control model, and a dual-interface experience that combined the traditional UI with conversational interaction. The integration did not require rewiring core infrastructure or rewriting workflows.



Key Results

With Adopt AI integrated:

- Users gained a co-pilot capability that answers procurement queries in context and within their existing authorization boundaries.
- Spendflo accelerated development of this feature without diverting significant engineering resources from the product roadmap.
- Adoption signals indicate reduced friction and improved engagement because customers can access data and complete actions without navigating multiple dashboards.



Engineering delivered the solution in a matter of days, maintaining governance, compliance, and performance while aligning with Spendflo's long-term product strategy. This case contrasts with the chatbot era, where conversational interfaces provided responses but did not execute actions or integrate deeply with backend systems. In Spendflo's deployment, conversational access connects directly to procurement workflows and data, driving outcomes rather than just surfacing information.

Customer Example | doola

How Doola Accelerated Business Operations With Agentic Execution

About Doola

Doola is a platform that supports entrepreneurs in starting, operating, and scaling U.S. businesses, including formation, compliance, taxes, and bookkeeping. Customers range from solo founders to global e-commerce operators managing complex compliance and financial workflows.

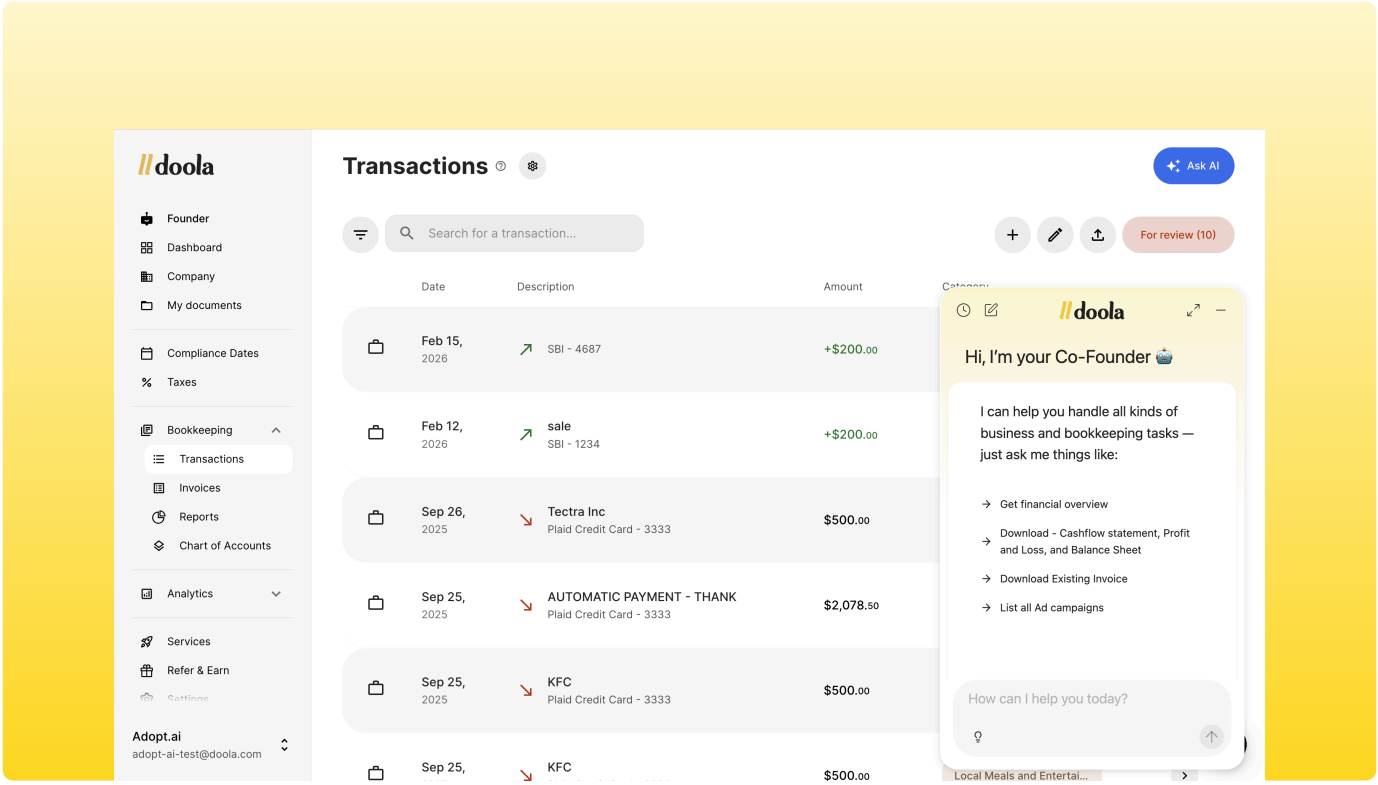
The Challenge

Entrepreneurs increasingly expect immediate, conversational access to operational data and actions rather than navigating complex dashboards and multiple screens. At the same time, Doola's product priorities made building and scaling artificial intelligence internally difficult without diverting scarce engineering capacity from core features.

Solution And Approach

Doola partnered with Adopt AI to introduce agentic workflows that could both retrieve enterprise information and execute actions directly within the existing platform. Instead of redesigning interfaces or rebuilding backend systems, the integration used existing APIs and infrastructure to embed context-aware, execution-capable agent flows. This allowed agents to operate within Doola's compliance-sensitive environments while maintaining accuracy and auditability.

The integration emphasized direct API-based execution and human-in-the-loop collaboration patterns. Agents were configured to handle document access, generate financial summaries, and support routine operational tasks without requiring Doola to rewrite workflows or restructure governance controls.



Key Outcomes

With Adopt AI integrated:

- Accelerated Product Delivery: Agentic workflows were deployed without burdening the internal engineering team, enabling faster rollout of new capabilities.
- Improved Customer Experience: Customers gained more natural, conversational access to business information and workflows, reducing friction in completing operational tasks.
- Engineering Leverage: Doola extended advanced execution capabilities without hiring specialized talent or slowing the product roadmap, building on existing infrastructure with partner support.

This example show how agents transition from exploratory tools to production systems that execute real work within established platforms. Unlike one-off conversational assistants, the integration enables agents that act securely and reliably in compliance-sensitive domains, bridging discovery and execution without reengineering core systems.

Voices From The Field

This section presents verifiable practitioner insights from real enterprises with documented agent deployments. Each quote is sourced from published customer experiences or executive statements directly tied to operational use, not hypotheticals.

Practitioner Insights

Across real enterprise deployments, we're seeing applications move away from UI-heavy workflows toward headless, terminal-driven execution. That shift does introduce a learning curve for product managers, who have historically relied on visual interfaces. But in practice, it's necessary. Agents that operate natively without UIs execute faster, scale more reliably, and avoid the cost and fragility of forcing machine workflows through human-designed interfaces."

Anirudh Badam Chief AI Officer @ Adopt AI

Platform Leader

"As a business handling high volumes of insurance claims, every minute we save counts. The structured framework brings order to what can feel like chaos, offering clarity on scaling agentic capabilities," said Bentubo in a customer testimonial about enterprise agent rollout. This reflects how defining precise deployment scope and scaling pathways reduced waste and confusion for her organization's claims processing workflows.

Kelly Bentubo Director of Architecture @ Alpine Intel (Insurance)

Enterprise IT Strategist

Shibani Ahuja emphasizes that scaling agent systems "requires a thoughtful, phased approach" with a clear maturity roadmap. Her guidance underscores that without defined stages for adoption and governance, organizations can create uneven or brittle automation landscapes.

Shibani Ahuja @ Salesforce

CTO & Technology Strategist

“LLMs aren’t reasoning machines, they’re just text prediction machines,” observes Schuerman. He cautions enterprises to “really anchor what your agents are doing in business processes” and build predictable, auditable workflows. In other words, start with well-defined goals and let the agent fill in the steps, rather than expecting it to magically reason.

Don Schuerman @ Pega

Executive Prediction

Salesforce field reports note that while agent adoption is growing rapidly, many leaders still hesitate to implement full autonomous agent strategies because “trust in data has become the number one bottleneck,” showing that operational confidence, not technology alone, determines success in real deployments.

Salesforce CIOs (Industry Study)

What They Learned

Clarity Is More Important Than Novel Technology

Teams discovered that agents only provide value when goals and constraints are specific and auditable. Alpine Intel’s experience, where a structured approach distilled chaos into measurable phases, shows why precision matters more than generative flair in enterprise contexts.

Governance Drives Reliability

Enterprise IT leaders learned that scaling without governance results in unreliable outcomes. Salesforce executives and CIO study data to make it clear that companies that align agent actions with data quality, access control, and audit frameworks see fewer failures and higher trust.

Incremental Rollouts Reveal Hidden Complexity

Organizations that attempted wide rollout without pilot staging encountered integration friction

and data mismatches. The pattern repeated in post-deployment surveys; trusted workflows start in narrowly scoped pilots, then expand once confidence metrics stabilize.

What Surprised Them

Speed Of Operational Adoption

While some expected slow uptake, agent deployments moved faster in high-volume, repetitive areas such as claims intake and customer engagement than anticipated. Real- world usage data shows that well-scoped agent tasks can be put into production within weeks when data pipelines are clean and governance is enforced.

Data Trust Issues Overshadow Technical Barriers

Executives predicted that model quality would be the largest barrier to adoption, but many reported that doubts about data trust and integration outweigh concerns about agent reasoning fidelity. Ensuring data lineage and access governance remains the dominant early challenge.

What Failed

Lack Of Outcome Definition Led To Drift

Practitioners shared that initial agent projects faltered when success criteria were vague. Without criterion for completion or quality thresholds, agents generated outputs that superficially looked correct but produced inconsistent operational results. This pattern shows up in early adopter feedback before governance layers were added

Overconfidence In “Plug-And-Play” Deployments

Some enterprise teams assumed that agents would work well simply by turning on automation tools. Experience showed that without tailored workflows and boundary conditions, agents tended to take actions that lacked context or violated procedural norms, necessitating rollback and redesign.

What Worked

Phased Adoption With Guardrails

Companies that implemented agents gradually; moving from assisted to autonomous tasks with intermediate controls, significantly reduced surprise failures and built confidence across stakeholders.

Data-First Preparation

Teams that invested in data quality initiatives and integration layers before deployment reported smoother transitions and fewer operational escalations. Reliable data access and a shared reference of truth allowed agents to act with measurable accuracy.

Shared Governance Boards

Enterprises that created cross-functional governance councils (including compliance, IT, and business owners) saw fewer conflicts and faster approvals for high-impact agent actions.

What They'd Do Differently

Invest Earlier In Verification Frameworks

Practitioners recommended making investment in verification pipelines and audit logging a priority from day one rather than as an afterthought, ensuring traceable decisions and minimizing risk during scale-out.

Define Escalation Policies Upfront

Teams said they would specify escalation thresholds and human-in-the-loop criteria at the outset, so agents hand off to humans clearly and without ambiguity.

Tie Metrics To Business Outcomes Rather Than Activity Measures

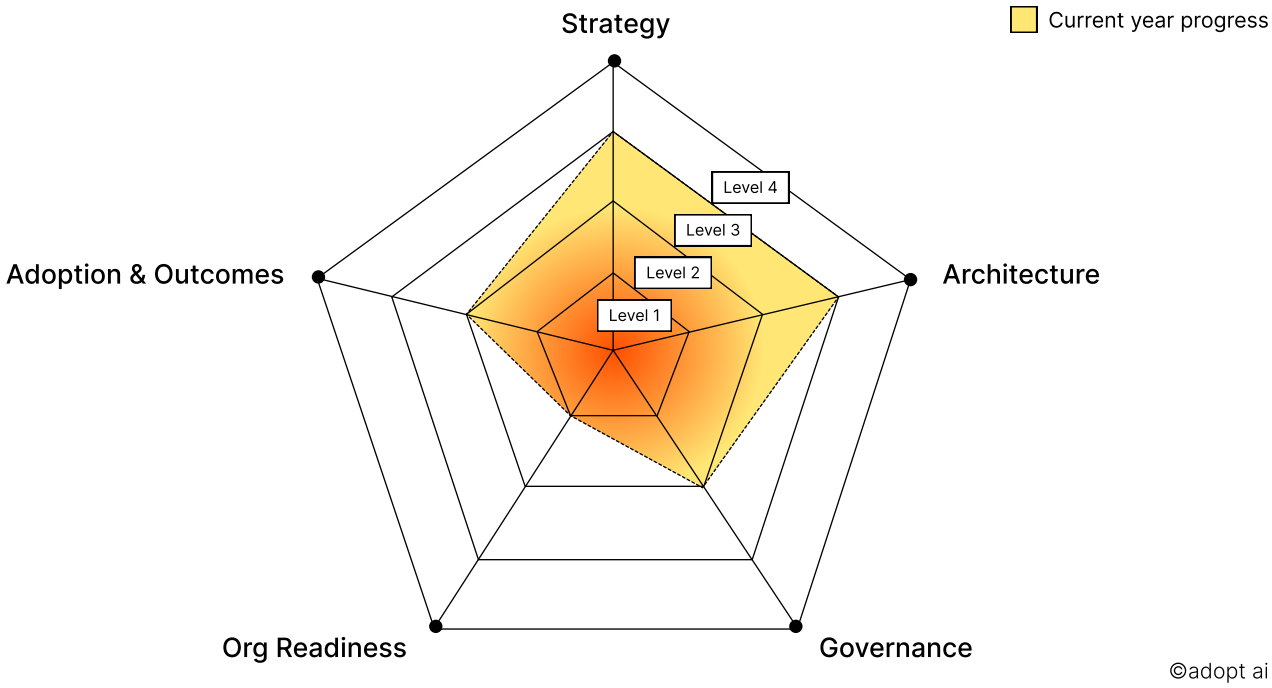
Instead of counting completed agent actions, future rollouts will focus on business outcomes (e.g., claim resolution cycle time, customer satisfaction improvements), which better reflect operational impact.

Research Methodology

This analysis synthesizes public research, vendor reports, and expert commentary to produce an up-to-date, unbiased view of agentic AI in the enterprise. We reviewed industry surveys (McKinsey, PwC, Gartner) and market forecasts, and we examined use cases and guidance from major AI and software vendors (Salesforce, SAP, Deloitte, IBM). We also drew on published practitioner interviews (e.g. CIO.com, trade press) and Adopt AI's own documentation to understand how organizations are implementing agents. All sources are cited to provide traceability. Where statistics or quotes are used, they come from these primary references. This approach ensures our conclusions reflect both aggregated data and on-the-ground insight.

Annual Agentic Scorecard

To gauge progress year-over-year, enterprises can use a scorecard with key dimensions. This scorecard is intended to expose capability gaps, not to validate activity or experimentation.



Each dimension is scored on a four-level scale that reflects operational reality rather than ambition. Organizations should score based on evidence from production systems, not pilot intent.

The Assessment

Strategy

This dimension evaluates whether agent adoption is deliberate, scoped, and leadership-backed.

Score Definitions

- Level 1 (Exploratory): Initiatives are isolated experiments; no executive sponsorship.
- Level 2 (Fragmented): A roadmap exists, but ownership is unclear and prioritization is weak.
- Level 3 (Targeted): Leadership has defined specific target workflows and success criteria.
- Level 4 (Integrated): Strategy is tied to business outcomes and reviewed alongside core metrics.

Validation Check: Can you name exactly which workflows are explicitly in scope for agents this year?

- Yes (Proceed)
- No (If No, your maturity is likely Level 1)

Self-Check Signals

If teams cannot name which workflows are explicitly in scope for agents this year, strategy maturity is low.

Architecture

This dimension measures whether agents can execute safely and reliably across systems.

Score Definitions

- Level 1 (Scripted): Agents rely on brittle UI automation or ad-hoc scripts; no shared infrastructure.
- Level 2 (API-Based): Agents use APIs, but lack a consistent orchestration or execution layer.
- Level 3 (Platformed): A shared execution platform exists with defined access boundaries.
- Level 4 (Orchestrated): Agents run in secured environments with state management and observability.

Validation Check: Can your agents be paused, restarted, or traced consistently by a central system?

- Yes (Proceed)
- No (If No, your maturity is incomplete)

Governance

This dimension evaluates control, traceability, and risk containment.

- Level 1 (Reactive): Monitoring is informal; issues are checked only after incidents occur.
- Level 2 (Logged): Basic logs exist, but there is no structured review or escalation process.
- Level 3 (Gated): Formal audit trails, approval gates, and escalation thresholds are enforced.
- Level 4 (Continuous): Governance is automated and reviewed as part of enterprise risk management.

Validation Check: Can compliance teams reconstruct why an agent made a specific decision?

- Yes (Proceed)
- No (If No, your maturity is likely Level 1 or 2).

Organizational Readiness

This dimension measures whether people and ownership models are prepared for agent operations.

Score Definitions

- Level 1 (Implicit): Oversight is handled ad hoc by technical teams.
- Level 2 (Fragmented): Ownership exists but is split across teams without clear accountability.
- Level 3 (Dedicated): Specific roles (e.g., AgentOps) supervise agent performance.
- Level 4 (Formalized): Lifecycle management (hire-to-retire) for agents is standard practice.

Validation Check: Is a specific team accountable if an agent fails silently?

- Yes (Proceed)
- No (If No, your readiness is low)

Adoption And Outcomes

This dimension measures whether agents are delivering measurable business impact.

Score Definitions

- Level 1 (Demo-Only): Agents are limited to pilots or demos with no production impact.
- Level 2 (Low-Risk): Agents operate in production but only cover narrow, low-risk workflows.
- Level 3 (Efficient): Agents handle end-to-end workflows with measurable efficiency gains.
- Level 4 (Transformative): Outcomes are tracked via business metrics and drive strategy.

Validation Check: Are you measuring business results (ROI/Speed) or just agent activity?

- Business Results (Proceed)
- Activity Only (If Activity Only, maturity is overstated).

This high-level scorecard, inspired by Gartner and our maturity model, helps companies identify gaps (e.g. strong pilots but weak governance) and track improvements each year.

Scoring Summary

Calculate Your Total Score (Sum of Levels 1–4): _____ / 20

- Score 5–8 (Experimental): You are in the "Prompt Experiment" phase. Focus on moving from UI automation to APIs and defining clear ownership.

- Score 9–14 (Operationalizing): You have "Task Agents" but lack scale. Focus on governance gates and orchestration.
- Score 15–20 (Agentic Enterprise): You are approaching "System of Record" status. Focus on lifecycle management and multi-agent ecosystems.

Next Review Date: _____ (Review annually to track progress against capability gaps)

Adopt AI Perspective

Adopt AI's platform is designed for exactly this agentic transition. Built for Application Use™, it automatically converts enterprise applications into actionable tools for AI agents. As Adopt's documentation explains, the era of point-and-click GUIs is ending, instead users want to "tell their apps what they need" in natural language. To achieve this, Adopt uses Zero-Shot API Ingestion (ZAPI) to capture all of an app's APIs and user flows within 24 hours, then makes each operation "LLM-ready" as a conversational action.

In practice, this means if an app can do X, the agent can learn to do X too. The platform then provides an agent builder where teams can compose prompts and guardrails around those actions. Crucially, Adopt emphasizes production readiness; it runs securely in your infrastructure (whether SaaS, VPC, or on-prem) and offers observability tools for tracking every agent invocation.

In this context, Adopt AI views agents not as vague assistants but as software components on top of real systems. Our focus is on enabling the higher maturity levels; by automating discovery and integration, helping enterprises move from "copilots" to goal- driven agents and systems of record more quickly.

What to Watch Next Year

Going forward, leaders should monitor how agentic AI standards, tools, and markets develop. Key signals include:

- Product announcements: Watch if major SaaS and enterprise software vendors (CRM, ERP, ITSM, etc.) start bundling built-in agents or agentic APIs. This will indicate mainstreaming.
- Open-source frameworks: Projects for agent orchestration (like OpenAI's tools or open stacks) will mature; surveys which become de facto standards.
- Regulation & ethics: Governments and industries may propose guidelines for autonomous AI. Companies should stay informed on emerging AI regulations that could affect agent use.
- Internal metrics: Track your own progress: e.g. percentage of processes with end-to-end automation, or service tickets handled by agents vs. humans. These internal scorecard numbers should improve each year.
- Talent shifts: Observe hiring and training, are roles like "AgentOps engineer" or "LLM developer" entering org charts? If so, adapt your org structure.
- Unexpected challenges: Keep an eye on security incidents or failures involving agents. Any high-profile "agent gone wrong" case will be a learning moment for the industry.

In summary, as agentic AI moves into production, leaders should continuously ask; Are our agents delivering value? Do we have the right oversight? How do we scale successful autopilots? By tracking these questions, enterprises can pivot faster when new AI innovations or shifts appear on the horizon. The evolution from copilot to autonomous agent is well underway, the next year will show which organizations can capitalize on it most effectively.

Sources: This report draws on the latest research and practice. For example, Deloitte provides adoption forecasts, Gartner lays out future timelines, SAP and Salesforce illustrate real use cases, IBM and Moveworks explain core concepts, and practitioner interviews (CIO.com) and PwC surveys offer on-the-ground insights. All source citations are provided for reference.

The Adopt AI Platform

End-to-end platform for building and operating enterprise agents.

Fragmented Tools & Integrations

- Connect to applications, data, and services without custom integration work
- Automatically generate safe, reusable system actions from existing workflows
- Enable cross-system execution without scripting or brittle mappings

Agent Builder & Interaction Components

- Build agents using natural language, configuration, or
- Use prebuilt UI components for chat, structured inputs, and history
- Support both user-activated and system-activated agent execution

Deployment, Scaffolding & Operational Tooling

- Full agent lifecycle support: testing, QA, deployment, monitoring, telemetry
- Embed agents into applications, workflows, and internal tools
- Engineering support (FDEs) for onboarding, scaling, and enterprise alignment

Enterprise Security & Transparent Economics

- SOC 2 Type II, ISO 27001, GDPR, client-side execution for sensitive operations
- Fine-grained permissions, audit trails, and policy enforcement
- Transparent, usage-aligned pricing with no hidden integration cost

Patented Technology

- PRN 3795: AI-BASED SYSTEM AND METHOD FOR AUTOMATED API DISCOVERY AND ACTION WORKFLOW GENERATION
- PRN 3796• SYSI-MAND METHOD FOR GEN-RALING AND EXECUNG APEDRIVEN ACHION PLAN.
- PRN 3797: SYSTEM AND METHOD FOR TRANSITIONING FROM LARGE LANGUAGE MODELS TO SPECIALIZED MACHINE LEARNING MODELS

To learn more

Visit Adopt.ai

Request Demo

+1 415 634 6253

info@adopt.ai

#1080, Plaza West, 3031 Tisch Way #110, San Jose, CA 95128