



ChatSee.ai

CHATSEE RESEARCH

State of Enterprise AI Failures 2026

How AI risk is shifting from model intelligence to system behavior. A cross-industry analysis of observed AI failure patterns across lifecycle stages, business functions, and agentic systems.

Published by ChatSee Research · 2026

AT A GLANCE

10,000+

observed AI failure events

150+

normalized failure
categories

10+

industry verticals

7

lifecycle stages

CORE THESIS

Enterprise AI is evolving from a model-quality problem into a systems-reliability problem. The most important failures increasingly occur around context, execution, access, escalation, and resolution — not only around hallucinations.

Contents

- Executive summary — The report's central argument and top five findings
- 1. About this study — Data sources, scope, methodology, and limitations
- 2. Understanding the enterprise AI lifecycle — Seven stages where AI systems can fail
- 3. The biggest failure families — Business-outcome view of the taxonomy
- 4. Where enterprise AI fails: industry view — Industry × lifecycle heat map and sector snapshots
- 5. Where business functions fail — Function × lifecycle heat map, customer-support vertical context, and operating patterns
- 6. The evolution of enterprise AI failures — Failure trends as systems become more agentic
- 7. Agent-type signals — Early directional evidence from coding, support, financial, and workflow agents
- 8. Emerging patterns — What the findings imply for runtime assurance
- Conclusion — What enterprise leaders should take away

Enterprise AI is failing in ways the industry is not yet measuring

For the past three years, the public conversation about AI risk has been dominated by hallucinations: models that fabricate facts, generate misleading answers, or respond confidently with incorrect information. That concern is real. But it is no longer the whole story.

Enterprise AI is moving from systems that answer questions to systems that retrieve information, invoke tools, execute workflows, generate code, analyze transactions, route cases, support customers, and participate in business operations. As AI systems become more operational, their failures are changing.

This report analyzes more than 10,000 observed AI failure events collected from public incident reports, open-source agent traces, Hugging Face datasets, AI interaction logs, production observations, and proprietary research conducted by ChatSee. The analysis spans more than 150 failure categories across more than 10 major industry verticals and multiple classes of AI systems, including conversational assistants, retrieval-augmented systems, coding agents, workflow agents, customer-service agents, and multi-agent applications.

CENTRAL FINDING

Enterprise AI is evolving from a model-quality problem into a systems-reliability problem.

The next generation of AI failures will not be defined only by wrong answers. It will increasingly be defined by systems that retrieve the wrong context, invoke the wrong tool, deny valid requests, fail to escalate, break workflows, or appear to provide service while never actually resolving the issue.

This shift has major implications for CIOs, CISOs, CAIOs, boards, product leaders, engineering teams, risk leaders, and AI governance teams. Model evaluations, prompt testing, red teaming, and output guardrails remain necessary. But they are not sufficient for AI systems that act inside enterprise workflows. The report identifies five findings that suggest the industry's current understanding of AI failure is incomplete.

Finding 1: Most enterprise AI failures are system failures, not model failures

The first surprise is that many enterprise AI failures do not originate in the model itself. They occur in the surrounding system: retrieval, context assembly, workflow orchestration, tool invocation, access control, escalation, response handling, and policy enforcement.

In many observed failures, the AI system appeared fluent and responsive. It understood enough of the request to continue the interaction. It generated plausible language. It may even have followed its immediate instruction. But the system still failed. It failed because the relevant context was not retrieved. Or because the correct workflow was not invoked. Or because a valid request was denied. Or because a high-risk case was not escalated. Or because the user received a polite response that did not actually resolve the issue.

One of the clearest signals in the data is that resolution and escalation failures were the largest mapped failure family, representing 31.1% of observed failures. That matters because these failures often look less dramatic than hallucinations. The AI is not obviously saying something bizarre. It is simply failing to move the issue to completion.

What is surprising: The industry has treated hallucination as the central AI failure mode. The data suggests that the largest enterprise risk may be more operational: systems that appear responsive but fail to resolve, escalate, or complete the work.

Why it matters: A better model alone will not fix failures in retrieval, workflow execution, entitlement logic, escalation, or resolution. Enterprises need to evaluate the behavior of the full AI system, not just the language quality of the model.

Finding 2: Different industries have distinct AI failure signatures

The second surprise is how strongly failure patterns vary by industry. Financial-services failures concentrate around response, escalation, governance, and resolution. Technology and telecom failures show much more execution and entitlement concentration. Healthcare and insurance failures are more sensitive to context integrity, stale information, policy interpretation, and patient or member guidance.

The pattern is not subtle. In the industry heat map, Technology & IT and Telecommunications & Digital Media showed some of the highest execution-failure concentrations, while Financial Services showed a much heavier response and governance profile.

The implication is more important than the individual numbers: the same AI architecture creates different risks depending on the industry in which it is deployed. A wrong or incomplete answer in retail may create a return, refund, or brand-trust problem. A similar failure in financial services may create fraud, conduct, compliance, or customer-harm exposure. A similar failure in healthcare may affect benefits, coverage, patient guidance, or care navigation.

What is surprising: AI failures are not distributed uniformly across industries. The failure signature changes with the industry's data architecture, regulatory context, operating model, entitlement structure, backend systems, and consequence model.

Why it matters: Generic AI controls are unlikely to be sufficient. A healthcare assistant, financial-services agent, telecom support bot, and coding agent require different assurance priorities because the failures that matter most are different.

Finding 3: The same AI function fails differently by industry

The third finding is the most important refinement to the industry-versus-function debate. Business function matters, but vertical context sharpens the failure pattern.

The report tested this directly by isolating customer-facing support and service records and recalculating lifecycle distribution by vertical. The result was clear: customer support does not have one universal failure profile.

Financial-services support was response and governance heavy. Food delivery and rideshare support leaned more toward policy, scoping, and exception handling. Telecom support showed governance and reasoning pressure tied to subscriptions, account state, provisioning, and entitlements. Travel support showed more context and reasoning exposure

because disruptions, refunds, itinerary changes, and rebooking workflows depend heavily on situational context.

In other words, “customer support AI” is not one risk category. The same function develops different hotspots depending on the surrounding industry context: the data it must retrieve, the policies it must apply, the workflows it must trigger, the systems it must access, and the consequences of getting the answer wrong.

What is surprising: Function-specific AI systems may share similar user experiences, but their failure modes diverge sharply once deployed into different verticals. A support agent in banking does not fail like a support agent in telecom, travel, retail, logistics, or healthcare.

Why it matters: Native platform analytics may show how an agent performs within a workflow, but enterprises also need vertical-aware failure intelligence: which lifecycle stages and failure hotspots matter most in this specific industry context. The finding strengthens the case for independent runtime assurance that can normalize failures across platforms while accounting for industry-specific risk.

Finding 4: Operational failures are growing faster than information failures

The fourth finding is about direction of travel. Hallucinations remain important, but the faster-growing failure categories are increasingly operational: execution failures, workflow breakdowns, escalation misses, resolution loops, tool-use failures, and action failures.

This pattern is consistent with the broader shift in enterprise AI architecture. The first wave of AI deployments generated text. The second wave retrieved information. The current wave increasingly invokes tools, executes workflows, writes code, updates systems, and coordinates actions across applications.

Each architectural step expands the failure surface. A chatbot can give a wrong answer. A retrieval system can answer from the wrong context. A tool-using agent can call the wrong API. A coding agent can break a build. A workflow agent can fail silently halfway through a process. A multi-agent system can lose ownership of resolution.

The time-based analysis reinforces this shift. In the observed failure mix, action and execution failures increased by roughly 62% relative to the Q2 2024 baseline, while hallucination-related failures declined by roughly 7% over the same period. This should not be read as a claim about total market incident volume. The important point is directional: as enterprises move from chatbots to agents, the observed failure mix is shifting toward systems that fail while performing work. The signal is not that hallucinations are solved. The signal is that operational failures are growing faster.

What is surprising: The fastest-growing risk is not simply that AI systems say incorrect things. It is that AI systems increasingly fail while trying to do things.

Why it matters: Controls designed for the chatbot era will not be enough for the agentic era. Enterprises need to monitor execution, state, workflow completion, escalation behavior, and recurrence — not just unsafe or inaccurate text.

Finding 5: The future of AI failures will look different than the past

The final finding is the strategic conclusion of the report. AI failure patterns are changing because AI systems themselves are changing.

In the chatbot era, the dominant concern was hallucination. In the retrieval era, context integrity became critical. In the tool-using agent era, execution reliability becomes central. In the coding-agent era, repository context, tests, dependencies, and deployment workflows become failure surfaces. In the multi-agent era, coordination, handoff, escalation, and resolution become first-class reliability challenges.

These risks do not replace one another. They accumulate. Future AI systems will still hallucinate. They will also retrieve the wrong context, misapply policy, invoke tools incorrectly, fail to escalate, deny valid requests, break workflows, and lose track of resolution across agent boundaries.

The agent-type analysis provides an early signal of this future. Coding agents showed a more distributed failure profile than traditional support workflows, while workflow and operations agents were more heavily action-oriented. Financial-services agents were far more resolution-heavy. These patterns are directional rather than exhaustive, but they point to the same conclusion: as agent types specialize, failure patterns specialize as well.

What is surprising: The industry is still largely debating yesterday's AI failure problem while deploying systems that create tomorrow's AI failure problem.

Why it matters: Organizations that focus only on model behavior will miss failures that emerge from the system around the model. The next phase of AI assurance must evaluate the complete runtime behavior of AI systems across context, reasoning, execution, governance, and resolution.

The larger implication

The evidence in this report points to a broader shift in enterprise AI risk. The first generation of AI failures was defined by incorrect answers. The next generation will increasingly be defined by systems that fail while participating in business operations. That changes the assurance problem.

The central question is no longer only: Can the model answer correctly? It is also: Can the system retrieve the right context, determine the right scope, take the right action, escalate at the right time, and bring the work to resolution?

Enterprises that understand this shift will be better positioned to deploy AI safely, govern it effectively, and improve it continuously. The future of enterprise AI will not be defined simply by who deploys the most agents. It will be defined by who understands how those agents fail — and who builds the systems required to make them reliable.

Top failure families across enterprise AI

Distribution of observed failures by business-outcome family.

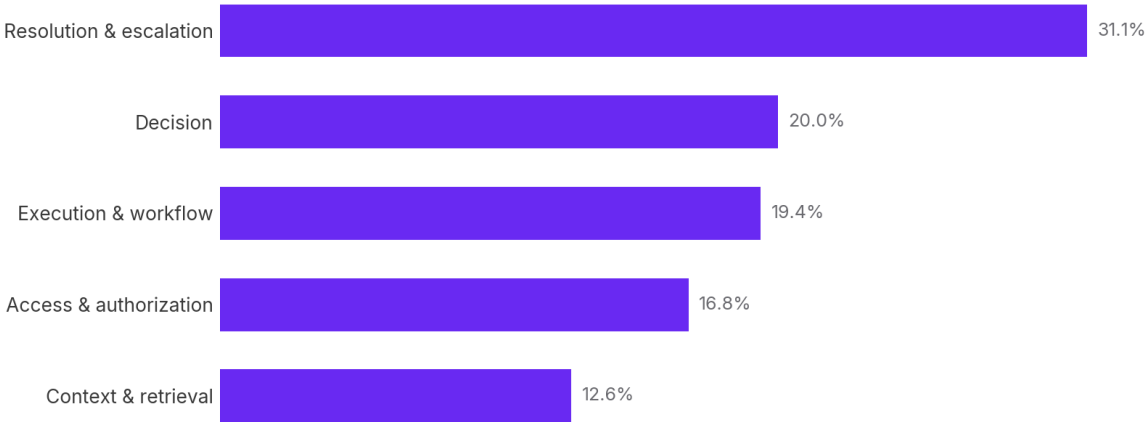


Figure 1. Failure families are grouped by business outcome. Shares represent observed failure distribution within the mapped corpus. Methodology note: percentages and indices in this report describe observed failure distribution within the analyzed corpus; they should not be interpreted as absolute market incident rates.

SECTION 1

About this study

This report is designed to map how enterprise AI systems fail in real-world use. It is not a benchmark of foundation models, a ranking of vendors, or a census of every AI incident. Instead, it is a structured analysis of observed AI failures across industries, business functions, agent types, and lifecycle stages.

The analysis incorporates more than 10,000 observed AI failure events collected between 2023 and 2026. Sources include public incident reports, community-reported failures, open-source agent traces, Hugging Face datasets, AI interaction logs, production observations, and proprietary research conducted by ChatSee.

The goal is to understand repeatable patterns. In enterprise AI, isolated anecdotes are useful but insufficient. A single hallucination may be interesting. A repeated pattern of escalation failure across financial services, customer support, healthcare, and logistics is more significant. It suggests a structural reliability problem.

The findings should be read as a map of emerging AI failure patterns, not as a definitive measurement of total AI failure rates across the market. The report identifies directional patterns in how AI systems appear to fail as they move from answer generation toward operational participation.

What the report measures

- Lifecycle stage — where in the AI system the failure occurred.
- Business outcome — what consequence the failure created.
- Industry vertical — which sectors exhibit distinct failure signatures.
- Business function — where enterprise work concentrates different risks.
- Agent type — how coding, support, financial, and workflow agents differ.
- Trend direction — which failure categories appear more prominent over time.

What the report does not claim

- It does not claim that every failure observed in public data represents a production enterprise incident.
- It does not claim that observed discussion volume equals actual incident volume.
- It does not claim that any individual model, product, or vendor is uniquely responsible for a given failure pattern.
- It does not claim causation between specific product releases and observed failure trends.

Understanding the enterprise AI lifecycle

Modern enterprise AI systems are not simply models generating text. They are increasingly runtime systems composed of multiple stages: interpretation, retrieval, scoping, reasoning, execution, response, and governance. Each stage can fail independently, and failures can propagate downstream.

A system that misunderstands intent may retrieve the wrong context. A system that retrieves incomplete context may reason from bad evidence. A system that reasons correctly may still fail during execution. A system that executes correctly may still fail to escalate a high-risk exception. This is why final-answer evaluation alone is insufficient. To understand enterprise AI failures, we need to understand where in the lifecycle they occur.

Lifecycle phase	How failures appear
Understanding	The AI system interprets the user's intent, entities, constraints, and desired outcome. Failures include misidentifying the task, ignoring constraints, or treating a complex request as a simple one.
Input processing & retrieval	The system gathers relevant context from documents, records, logs, policies, repositories, or prior interactions. Failures include stale context, missing evidence, or incorrect grounding.
Scoping & entitlements	The system determines whether the request is valid, supported, or allowed. Failures include wrongful denial, entitlement mismatch, and blocked legitimate workflows.
Reasoning	The system evaluates evidence and determines what should happen. Failures include hallucinations, logic errors, misclassification, policy misapplication, and unsupported conclusions.
Execution	The system invokes tools, APIs, workflows, repositories, or downstream systems. Failures include missing credentials, invalid parameters, build errors, integration breakdowns, and partial completion.
Response	The system communicates an outcome, explanation, recommendation, or next step. Failures include generic non-resolving responses, repetitive loops, and incomplete guidance.
Policy & governance	The system applies enterprise rules, escalation criteria, safety controls, and compliance logic. Failures include failed escalation and inconsistent policy enforcement.

Observed failures across the enterprise AI lifecycle

Distribution across lifecycle phases; failures emerge before, during, and after model reasoning.

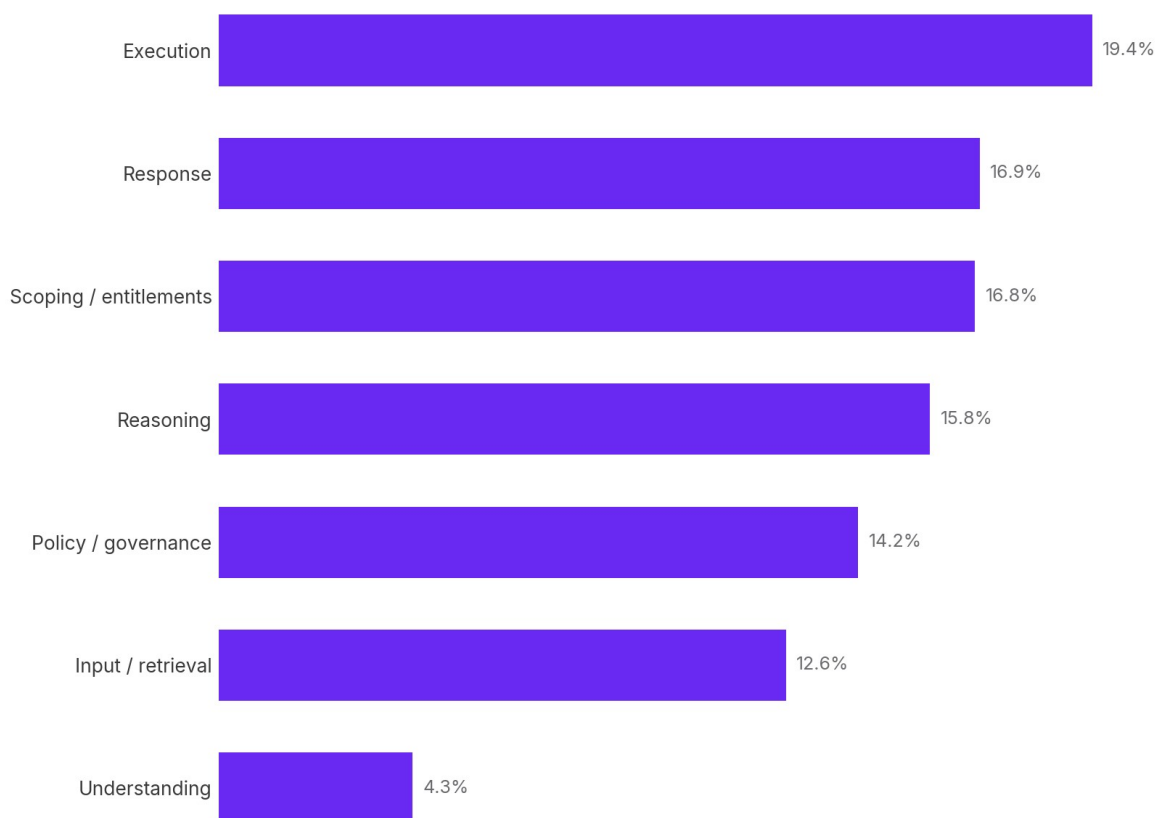


Figure 2. Observed failures are distributed across the lifecycle rather than concentrated only in model reasoning.

SECTION 3

The biggest failure families in enterprise AI

Raw error codes are useful for diagnosis, but they are not how executives, boards, regulators, or journalists understand risk. For this report, the error taxonomy is rolled up into business-outcome failure families. This framing answers a more important question: what business outcome was affected?

Resolution & escalation failures

The AI system does not bring the issue to completion. The user may receive responses while no accountable resolution occurs. Common patterns include failed escalation, dead-end service loops, improper handoffs, and repeated troubleshooting without progress.

Representative example: A fraud-investigation assistant repeatedly collects information from a customer but fails to escalate the case after suspicious-account criteria are met.

Decision failures

The AI system reaches the wrong conclusion. This includes hallucinations but also broader decision-quality failures such as logic errors, misclassification, unsupported conclusions, and policy misapplication.

Representative example: A customer-support agent misclassifies a billing dispute as a technical-support issue, sending the user into the wrong resolution path.

Execution & workflow failures

The AI system attempts to execute work and fails. Common patterns include tool invocation errors, workflow execution failures, missing credentials, API errors, and build or deployment failures.

Representative example: A coding agent modifies files and passes local checks, but introduces a dependency conflict that breaks downstream CI/CD execution.

Access & authorization failures

The AI system cannot access required information, accesses the wrong information, or incorrectly denies a valid request. These failures can create customer harm and operational rework even when they are not data leaks.

Representative example: A wealth-management assistant denies an advisor access to information the advisor is entitled to receive, blocking a legitimate client-service workflow.

Context & retrieval failures

The AI system acts on incomplete, stale, incorrect, or missing context. These failures often produce plausible answers because the language is fluent but the evidence base is wrong.

Representative example: A healthcare assistant retrieves outdated coverage information and provides guidance inconsistent with current benefits.

Where enterprise AI fails: industry view

The industry-level analysis shows that AI failure patterns vary by vertical. This matters because AI systems are not deployed into generic environments. They operate inside specific business processes, regulatory contexts, customer expectations, and operational workflows.

A failure in a retail chatbot may create an incorrect product recommendation. A failure in a financial-services assistant may prevent escalation of suspicious activity. A failure in a healthcare assistant may produce guidance based on stale or incomplete information. The same underlying AI capability can create very different business risks depending on where it is deployed.

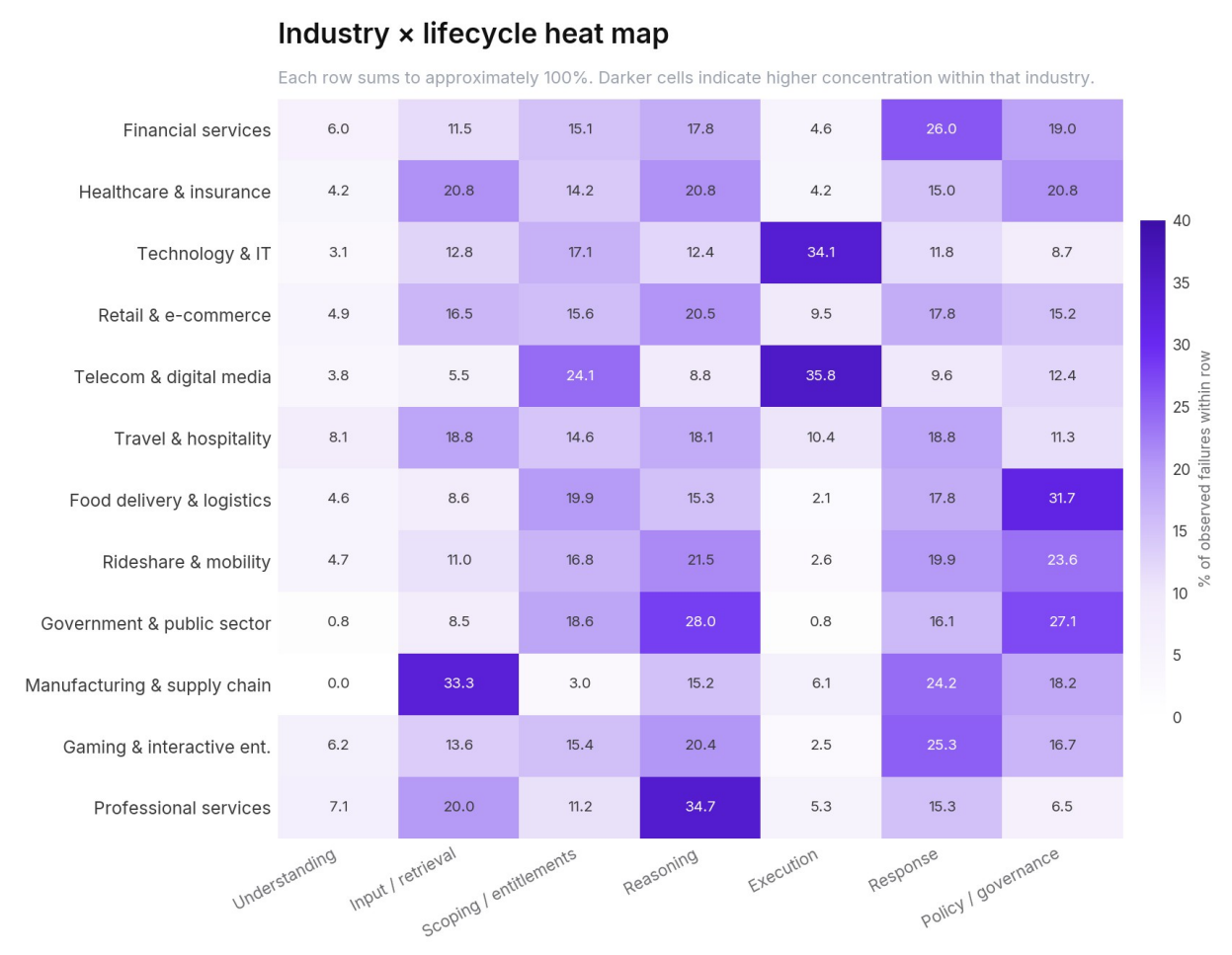


Figure 3. Industry × lifecycle heat map. Each row sums to approximately 100%; darker cells indicate higher concentration within that industry. The table shows distribution within each industry, not volume by industry — a dark cell means that phase accounts for a larger share of observed failures in that industry.

Industry	Signature pattern	Representative example	Why it matters
Financial services	Resolution and escalation failure	A banking or brokerage support assistant repeatedly	In financial services, failure to resolve can be

Industry	Signature pattern	Representative example	Why it matters
		engages with a customer reporting suspicious account activity but fails to escalate the case to a fraud or account-protection workflow.	more consequential than a wrong answer. Escalation, auditability, and accountable review are central to operational risk management.
Healthcare & insurance	Context integrity and policy-sensitive guidance	A patient-support assistant provides guidance based on outdated formulary or benefit information.	In healthcare, context errors can affect access, trust, coverage, and care navigation.
Technology & software engineering	Execution and workflow reliability	A coding agent generates a plausible fix but breaks downstream CI/CD execution due to an unhandled dependency or environment mismatch.	Coding-agent failures often look like ordinary software delivery failures introduced by autonomous systems.
Customer support	Decision and resolution failure	A telecom support assistant repeatedly asks a user to reboot equipment even though prior diagnostics indicate a provisioning issue requiring human intervention.	Support AI can reduce apparent ticket volume while increasing unresolved customer frustration.
Retail & e-commerce	Persuasive but incorrect commercial guidance	A commerce assistant recommends a product combination that appears plausible but is incompatible with the user's stated requirement.	Commerce AI failures directly affect conversion, returns, trust, and brand reputation.
Travel & hospitality	Exception-handling loops	A travel assistant repeatedly provides generic booking guidance while failing to resolve a disrupted itinerary requiring exception handling.	Travel failures occur at moments of high customer stress and time sensitivity.
Telecom & digital media	Account-entitlement and execution traps	A support assistant correctly understands a streaming-access complaint but cannot resolve the underlying entitlement mismatch.	The AI may understand the complaint but fail because account state and backend systems do not align.
Public sector	Authority-risk amplification	A government assistant provides incorrect procedural guidance about a benefits, permit, or filing process.	When public-sector AI gives incorrect guidance, the harm may be amplified by institutional trust.

Where business functions fail

Industry tells one story. Business function tells another. The same industry may deploy AI across customer service, fraud, compliance, software engineering, sales, operations, internal support, and back-office workflows. Each function creates a different failure profile.

Business function is one of the report's top analytical lenses. Failure patterns are determined not only by the model or the industry, but by the work the AI system is asked to perform. The customer-support analysis later in this section adds an important nuance: the same function can develop different failure hotspots depending on vertical context.

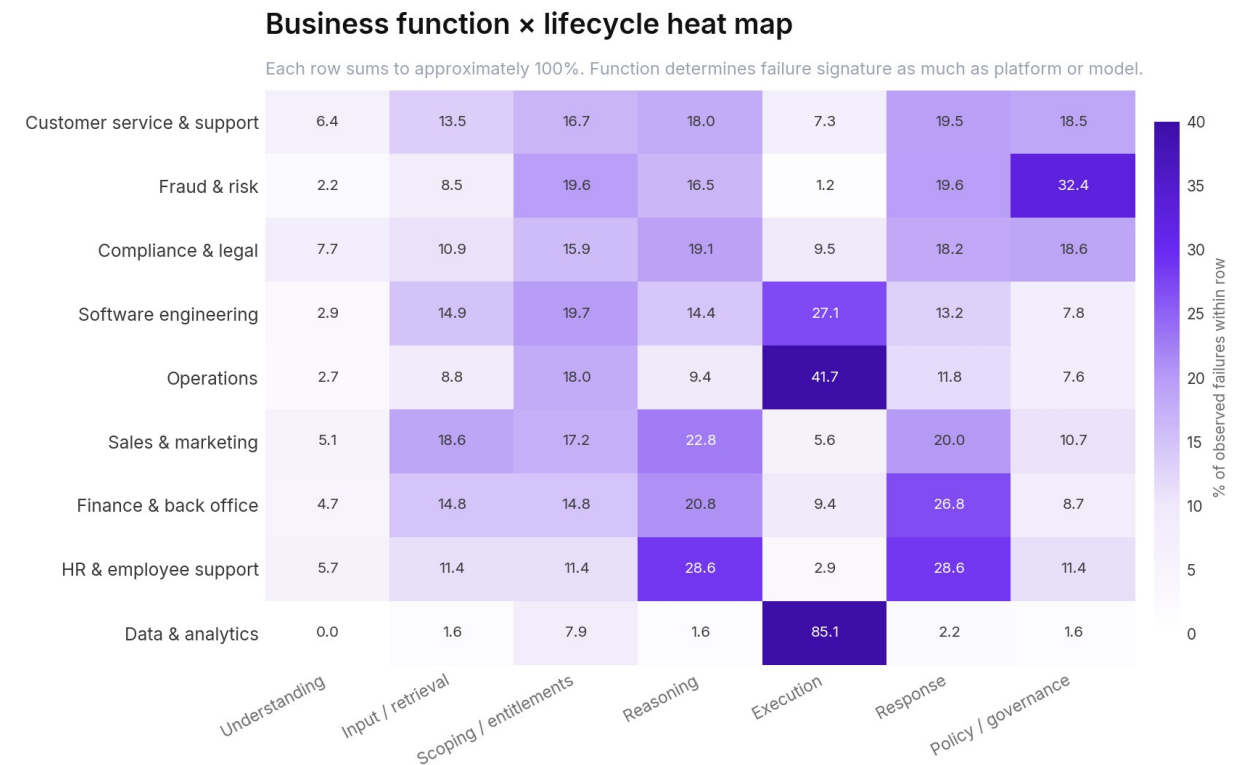


Figure 4. Business function × lifecycle heat map. Each row sums to approximately 100%; darker cells indicate higher concentration within that function. Function-level patterns show where risk materializes in enterprise workflows — the same model can fail differently when used in customer service, fraud, engineering, HR, or analytics.

Business function	Signature failure pattern
Customer service & support	Response, resolution, and escalation; the appearance of service without actual resolution.
Fraud & risk	Policy, escalation, and response; the critical question is often whether automation should stop and escalate.
Compliance & legal	Reasoning, response, and governance; incorrect interpretation of policy or obligation is the core risk.

Business function	Signature failure pattern
Software engineering	Execution, workflow, scoping, and repository context; the AI operates inside real development environments.
Operations	Execution; the AI may understand the task but fail to complete it across tools and systems.
Sales & marketing	Reasoning, response, and input context; persuasive but unsupported claims create external risk.
Finance & back office	Response, reasoning, and context; approvals, policy interpretation, and record accuracy are central.
HR & employee support	Reasoning and response; nuanced policy-sensitive employee situations require context and careful guidance.
Data & analytics	Execution; analytics agents fail when querying, joining, transforming, or generating outputs from enterprise data.

Customer support across verticals: function does not erase context

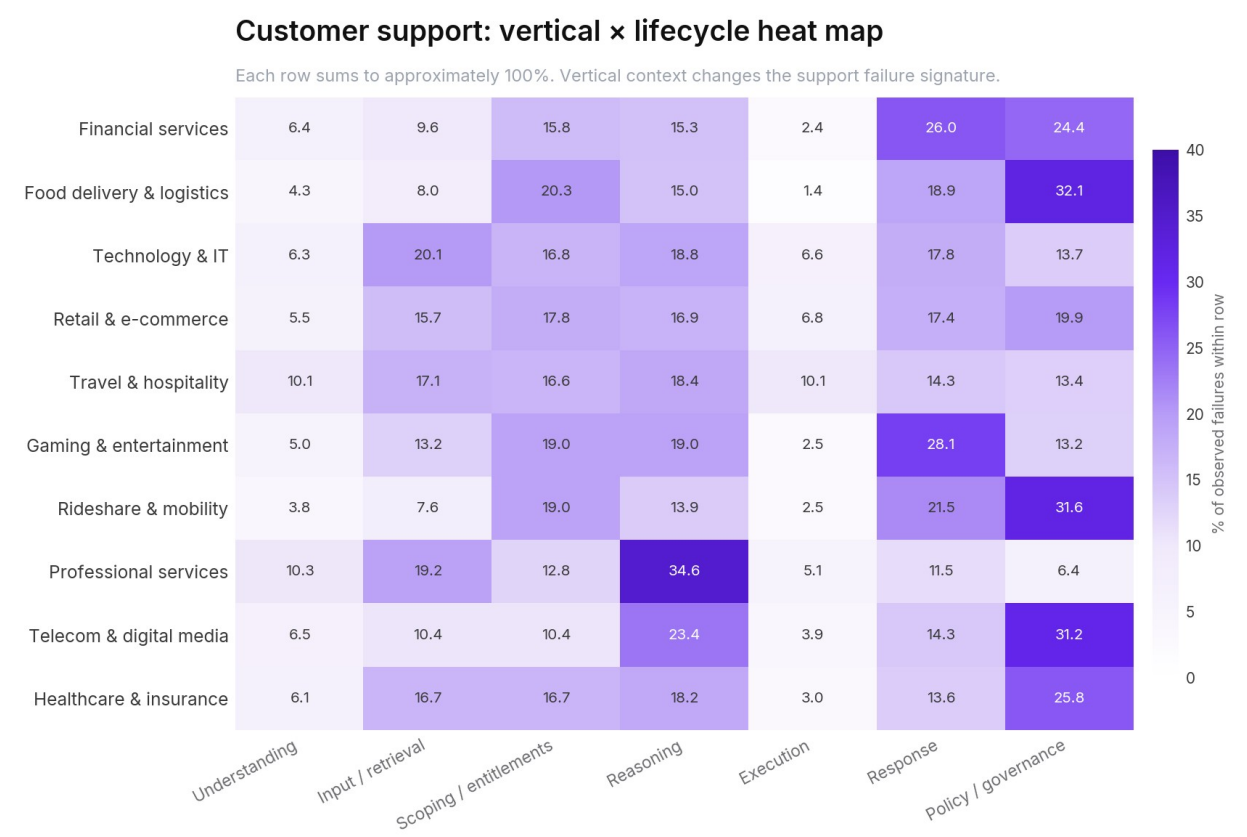


Figure 5. Customer support vertical × lifecycle heat map. Each row sums to approximately 100%; darker cells indicate higher concentration within that vertical's customer-support subset. This view isolates customer-facing support records — it should be interpreted as vertical-specific failure mix, not total support volume by vertical.

To test whether function-level patterns hide vertical-specific signatures, ChatSee isolated customer-facing support and service records and recalculated lifecycle distribution by vertical. The result is clear: customer support does not have a single failure profile.

Banking, telecom, travel, retail, healthcare, delivery, and technology support systems may all perform the broad function of customer service. But they fail at different lifecycle hotspots because the surrounding data, policy, entitlement, escalation, and operational systems differ.

What the pattern means

Business function provides a useful baseline. Customer service generally elevates response, resolution, and escalation risk. But vertical context sharpens the signature. Financial-services support is response and governance heavy. Food delivery and rideshare support are policy and scoping heavy. Technology and travel support show more context and reasoning exposure. Telecom support shows governance and reasoning risk tied to entitlement, provisioning, and account-state issues.

The implication is that a function-specific AI system should not be evaluated only through generic support metrics. A customer-service agent needs a different assurance focus in banking, healthcare, telecom, travel, retail, and logistics because the underlying failure hotspots are different.

COMPLEMENTARY LENSES

Function describes the work; vertical context defines which failures are most likely to matter.

The evolution of enterprise AI failures

The analysis suggests that the enterprise AI failure landscape is changing. Early AI deployments were heavily associated with answer-quality failures: hallucinations, incorrect facts, misleading summaries, and unsupported claims. Those failures remain important. But as AI systems become more operational, new failure classes become more prominent: execution failures, escalation failures, workflow breakdowns, tool-use errors, and resolution dead ends.

AI architecture	Primary failure concern
Chatbot	Hallucination and wrong answer
RAG system	Retrieval and context integrity
Tool-using agent	Execution and workflow reliability
Coding agent	Repository context, test, build, and deployment reliability
Multi-agent system	Coordination, handoff, escalation, and resolution

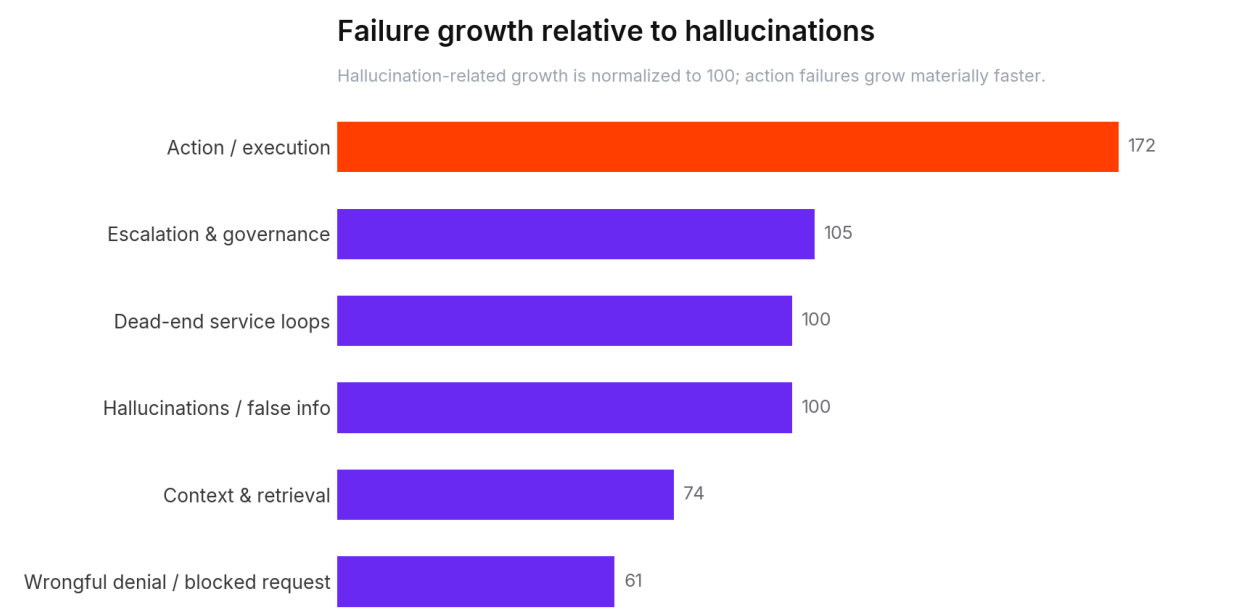


Figure 6. Relative growth analysis. Hallucination-related growth is normalized to 100; action failures show materially faster growth in the observed failure mix.

Monthly cumulative failure index

Q2 2024 = 100 for each category; index reflects observed failure mix, not absolute market incident rate.

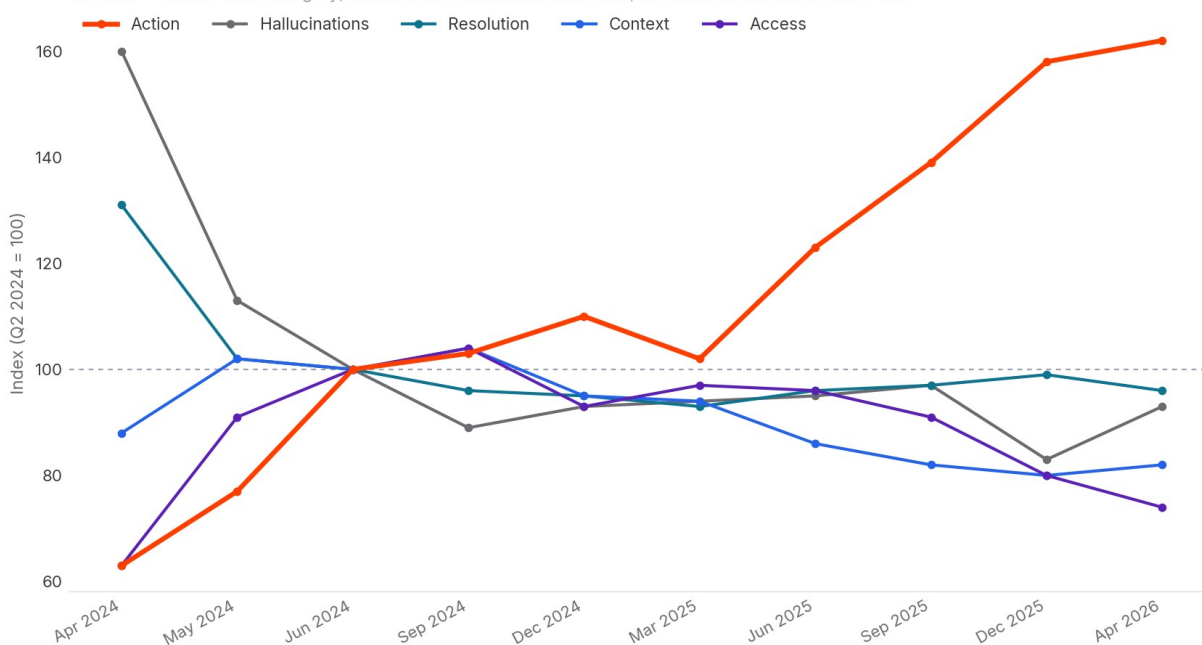


Figure 7. Monthly cumulative failure index. Q2 2024 = 100 for each category; the clearest directional pattern is the rise of action failures.

The trend charts should be interpreted as directional signals in observed failure mix, not as claims about absolute market incident rates. The important signal is that execution and action-related failures are becoming more prominent as AI systems move closer to enterprise workflows.

Many enterprise AI controls were designed for the chatbot era. They focus on unsafe outputs, hallucinations, toxic content, prompt injection, data leakage, and policy-violating responses. These controls remain necessary. But they are insufficient for agents. An agent can pass an output filter and still invoke the wrong tool, proceed with incomplete context, fail to escalate, deny a valid request, enter a resolution loop, or break a workflow.

Agent-type signals

As AI systems become more specialized, their failure profiles diverge. This section is intentionally directional. It does not claim a comprehensive typology of all agent classes. Instead, it shows early evidence from four mapped groups: coding agents, customer-support agents, financial-services agents, and workflow or operations agents.

The purpose is to establish an emerging pattern that future editions of the State of Enterprise AI Failures series will examine in greater depth.

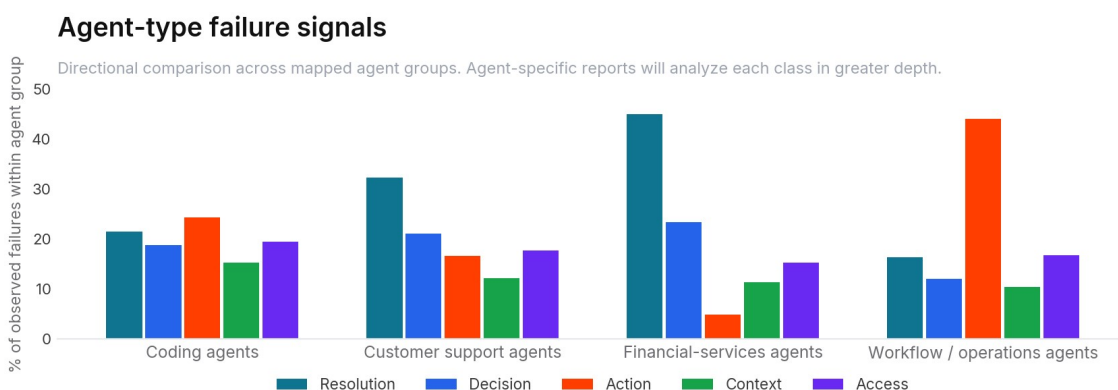


Figure 8. Agent-type signals. Different agent groups show different failure concentrations; future reports will analyze specific agent classes in depth.

Coding agents

Balanced failure distribution across decisioning, context, execution, resolution, and access. These systems must understand requests, retrieve repository context, reason about code, edit files, run tests, and determine whether a task is complete.

Customer-support agents

Stronger concentration in decision and resolution failures. These systems often fail not because they cannot generate language, but because they misclassify the issue, provide generic responses, or fail to escalate.

Financial-services agents

Strong concentration in resolution and escalation failures. Many valid outcomes require review, exception handling, auditability, and accountable human intervention.

Workflow / operations agents

Clearest concentration in action failures. These systems fail when they attempt to perform work across tools, records, APIs, permissions, and operational state.

Emerging patterns

The rise of resolution failures

Many AI systems now fail by appearing to help without actually resolving the issue. A user may receive repeated responses, instructions, apologies, or policy explanations while the actual problem remains unresolved. In conversational AI, a failed workflow can be hidden behind polite interaction.

The system failure problem

The model is only one part of the AI system. Enterprise AI systems depend on retrieval infrastructure, connectors, APIs, tooling, permissions, identity systems, workflow engines, human handoff, policy layers, monitoring, and state management. Failures in any of these layers can create AI failure.

Governance as runtime behavior

For AI agents, governance increasingly becomes runtime behavior. The system must decide whether to answer, refuse, escalate, authorize, review, or stop. These are dynamic decisions, not static checklist items.

Multi-agent coordination risk

As organizations move toward multi-agent systems, coordination becomes a first-class reliability concern. Multi-agent systems introduce handoff failures, conflicting outputs, planner/executor mismatch, repeated work, lost state, ownership gaps, and escalation gaps.

The need for runtime assurance

Pre-deployment evaluation remains necessary, but it is not sufficient. Modern AI systems require runtime assurance: detecting behavioral failures, classifying failures by lifecycle stage, tracking recurrence, measuring business impact, identifying escalation gaps, creating failure memory, and feeding remediation back into engineering, operations, and governance.

CONCLUSION

What enterprise leaders should take away

The first generation of enterprise AI failures was defined by incorrect answers. The next generation will increasingly be defined by systems that fail while taking action.

As AI systems retrieve information, call tools, execute workflows, coordinate across applications, and participate in business processes, reliability shifts from model intelligence toward system behavior.

This does not make hallucinations irrelevant. It places them in a broader context. The central question for enterprise AI is no longer only: can the model answer correctly? It is also: can the system retrieve the right context, determine the right scope, take the right action, escalate at the right time, and bring the work to resolution?

Organizations that understand this shift will be better positioned to deploy AI safely, govern it effectively, and improve it continuously.

BOTTOM LINE

The future of enterprise AI will not be defined simply by who deploys the most agents. It will be defined by who understands how those agents fail — and who builds the systems required to make them reliable.

Methodology notes

This report uses percentage distributions, relative indices, and normalized failure-family shares rather than raw failure counts. That choice is intentional: the objective is to compare failure signatures across lifecycle stages, industries, functions, and agent types without implying that the underlying corpus is a census of all AI failures in the market.

Heat maps represent distribution within each row. For example, a dark cell in the financial-services row means that a larger share of observed financial-services failures occurred in that lifecycle phase. It does not mean financial services had more total observed failures than another industry.

Customer-support vertical analysis isolates records mapped to customer-facing support and service workflows. Percentages are calculated within each vertical row so the comparison highlights failure signatures rather than support volume.

Trend indices represent changes in observed failure mix over time. They should be interpreted as directional indicators, not absolute incident-rate measurements.

Agent-type analysis is intentionally directional. The flagship 2026 report uses agent-type evidence to identify emerging patterns; future reports will examine specific agent classes in greater depth.