How To Trust AI Agents

Idea In Short

An agent can recommend the right refund and still pay it twice. Evaluation has to cover the whole workflow, from permissions and integrations to the human review around them, and authority should grow only as far as the evidence reaches. Five decisions sit with the executive sponsor of any agent deployment, covering how to define success for a workflow, how to test it under realistic conditions, how to verify permissions and approvals, how to use a pilot to set the level of autonomy, and how to keep approval conditional after launch. A refund-handling agent serves as the running example throughout, and a checklist at the end supports the final sign-off.

Agents have moved from drafting text to doing work. They look up orders, update customer relationship management [CRM] records, send messages and trigger payments. Each new action raises the price of a mistake, and a polished demonstration reveals little about how often mistakes occur.

A good demo settles very little

Gartner predicts that more than 40% of agentic AI projects will be canceled by the end of 2027, citing escalating costs, unclear business value and inadequate risk controls.1 Research on tool-using agents adds a sharper data point. The tau-bench benchmark places agents in retail customer-service tasks governed by company policy. When it was published in 2024, state-of-the-art function-calling agents succeeded on fewer than half the tasks, and their pass^8 score, the share of tasks solved on all eight repeated attempts, fell below 25% in retail.2 Models have improved since then, but the lesson holds. One impressive run says little about the tenth.

For executives and boards, the useful question is how much authority the available evidence supports.

Start from the action, not the answer

Take a customer-service agent that recommends refunds. It pulls the order, checks the return policy and explains why the customer qualifies. A manager reads the explanation and approves. Errors at this stage are cheap to catch.

Let the same agent issue the payment and the picture shifts. Has this order already been refunded? Does the payment tool confirm success? What happens when the request times out? A well-written explanation sheds little light on any of these.

AI agent evaluation should therefore tie observed performance to the authority the business plans to grant. Leaders can approve recommendations today, keep payments under review, and revisit wider autonomy once testing supports it.

Define the workflow and what counts as success

Begin with a process the operations team can describe step by step. List the inputs, decisions, connected systems, actions and final records, then name the owner and the people affected when errors occur.

For the refund agent, success means the right customer receives the right amount exactly once and the order record shows the outcome. Sorting actions into three groups helps.

1. Preparing a recommendation is permitted
2. Initiating a payment requires approval
3. Opening an unrelated customer account is off-limits

Measure the current process first, covering completion time, error types, review effort and cost per case. This baseline becomes the yardstick for every later comparison. The National Institute of Standards and Technology's AI Risk Management Framework asks organizations to map the context a system will operate in before they measure its performance,3 which in practice means writing acceptance criteria before looking at any results.

Weight those criteria by consequence. A clumsy explanation and an unauthorized payment call for different responses, and blending both into one quality score can bury the failure that should stop a launch.

Test the whole workflow, not just the reply

Build the test set from historical cases the business handled correctly, then add constructed exceptions. Ordinary requests belong in realistic proportions, alongside targeted tests of serious failure modes. Hold back a separate final set so repeated tuning cannot overfit to familiar examples.

Record the configuration under test, including model version, instructions, data sources, tools and permissions. A passing result then describes the system that actually ships.

Look past the final message. Check facts, instruction adherence, tool choice, the parameters passed and the business records left behind. Rerun selected cases, since the tau-bench results show how far a single pass can mislead. Include missing order numbers, conflicting policies, unavailable services and unusual requests. When information is insufficient, a correct escalation counts as success.

Partial execution deserves special attention. Suppose the payment service times out after receiving a refund request. The refund may have gone through despite the missing confirmation, and a blind retry could pay the customer twice. The surrounding integration should be able to check transaction status or block duplicates. If it can do neither, the agent should leave the case open and escalate it with the attempted action recorded.

Sort failures by severity and recoverability, and report how many cases were run and which conditions they covered. Zero critical failures in a finite sample still leaves room for uncertainty.

Check that permissions and approvals hold up

Written instructions form one layer of control, and a thin one. Review what the accounts and tools behind the agent are allowed to do, then attempt prohibited actions in an isolated test environment and confirm the system refuses them.

OWASP traces the risk of excessive agency to three causes, namely excessive functionality, excessive permissions and excessive autonomy. It recommends enforcing authorization in downstream systems instead of trusting the model to police itself.4 For the refund agent, permission to read order data stops short of exporting the customer database, and

permission to prepare a payment stops short of executing one.

Seed test inputs with hostile instructions, such as a customer message or retrieved document telling the agent to ignore refund limits or forward records elsewhere. Text the agent reads should never acquire the force of a command.

Examine vendor and integration data handling too, covering retention, storage, log access and contractual use of submitted information. Evaluation records often contain confidential data, so protect them as well.

Enforce approvals before execution. The reviewer should see the customer, amount, applicable policy and intended action. Test rejection, expired approvals, an absent reviewer and attempts to reuse one approval for a different transaction. Name a business owner who is accountable, and give employees the authority and a documented procedure to intervene.

Let the pilot set the level of autonomy

Run a limited pilot with a defined scope, review period and stopping conditions. Where practical, start in shadow mode. The system prepares proposed actions without executing them, and the team compares those proposals with what the existing process produced.

Move to supervised execution once the earlier evidence supports it. Cover normal volumes, busy periods and controlled failure tests, arranged so live customers stay protected from induced failures.

Track completion, correction effort, escalations, latency and cost per correctly completed case, counting model usage, integrations, monitoring, maintenance and human review. A small software bill means little when staff spend hours repairing the output.

The pilot should end in a scoped decision instead of a single yes or no. The table below shows one way to translate evidence into authority.

Evidence observed	Appropriate decision
Recommendations are useful, execution controls remain unverified	Keep the agent advisory
Outcomes meet criteria with manageable human review	Permit supervised execution

Evidence observed	Appropriate decision
Bounded actions meet criteria and verified controls limit consequences	Consider autonomy for those actions only
Serious control failures remain unresolved or total costs outweigh benefits	Pause deployment or redesign the workflow

These thresholds are suggestions. Each organization sets its own acceptance criteria and weighs the remaining uncertainty.

Keep critical failures visible. Strong average accuracy cannot offset a demonstrated ability to bypass payment approval, so fix the cause and retest the affected controls before widening access.

Keep approval conditional after launch

Approval should name the workflow, configuration, permissions and operating limits the evidence covers. Assign someone to watch failure patterns, intervention rates, costs and unresolved transactions.

Agree on suspension triggers in advance. Unauthorized access, duplicate payments and missing transaction records may justify immediate intervention, while a rising correction rate may call for investigation before any expansion.

Retest after changes to models, instructions, policies, integrations or permissions, scaling the effort to the change and its potential consequences. Keep records sufficient to reconstruct inputs, retrieved information, tool calls, approvals and outcomes, holding only the sensitive data this purpose requires.

Rehearse shutdown and recovery as well. Confirm who can disable the agent, what happens to work in progress, and how open cases return to staff. Some actions cannot be reversed, so recovery may involve reconciliation and corrective steps.

Before anyone signs off

The accountable owner should be able to confirm each of the following.

- The workflow, permitted actions, baseline and acceptance criteria are documented
- Representative cases, repeated runs and consequential failure scenarios have been tested
- Data handling, access restrictions and approval controls have been verified
- Tool failures and uncertain transaction outcomes have a defined response
- Pilot results include review effort, operating costs and unresolved errors
- Monitoring, intervention, recovery and change-triggered retesting have named owners

- 1Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027
- 2tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains
- 3AI Risk Management Framework
- 4LLM06:2025 Excessive Agency

Summary

Authority should track evidence, not enthusiasm. An agent can be ready to recommend refunds while still short of issuing them, and that finding counts as a useful result rather than a disappointing one. Repeated runs, failure-path tests and verified permissions tell a business far more than one successful demonstration ever could. Approval stays conditional, backed by named owners, suspension triggers and retesting after every material change to the model, the instructions, or the permissions behind it. In the end, readiness comes down to one question: whether the business can tolerate and control the consequences when a workflow fails.