EXECUTIVE ARTICLE / AI & GOVERNANCE

Before You Give an AI Agent Authority, Decide What It Must Escalate

A useful agent needs a clear boundary between making a recommendation and making a commitment.

Jennifer Oriwoh

Consider a hypothetical customer-service assistant handling a refund. It finds the order, reads the policy and prepares a sensible response. The demonstration looks convincing. Then someone connects it to the payment system.

The next question is harder than whether the assistant understands refunds. Can it issue one when the delivery record conflicts with the customer's account? Can it change the destination account? What happens if the payment service accepts the request but the confirmation never arrives?

Those are decisions about authority, evidence and recovery. A successful demonstration does not settle them. Before expanding an agent's access, a business should define the actions it can complete independently, the circumstances that require escalation and the person responsible for resolving an exception.

Separate capability from permission

The word agent can obscure an important design choice. In its 2024 engineering article, Anthropic distinguishes predefined workflows from systems in which a model directs its own steps and tool use. It also describes a tradeoff between flexibility, cost and latency. The article reflects a vendor's engineering experience; it is useful architectural guidance, rather than independent proof of a business return.

That distinction matters to the refund example. A fixed workflow can require an order lookup, an eligibility check and an approval before payment. A more flexible agent might decide which records to consult and when to act. Neither arrangement makes the payment decision harmless. Both need permissions enforced by the surrounding software.

An instruction telling the model to be careful is too weak a foundation for authority. The tool that issues refunds should enforce the permitted amount, eligible transactions and approved destination. Requests outside that scope should be rejected or routed for review, even when the agent supplies an eloquent justification.

Give escalation a destination

An escalation policy should name observable conditions. Conflicting records, missing evidence, a request to bypass a control and an uncertain payment outcome are more useful triggers than a general instruction to escalate difficult cases. Difficulty is subjective; an unresolved discrepancy can be identified and recorded.

The receiving person also needs a usable case. Sending an entire conversation with a request to investigate transfers the workload without organising it. A better handoff includes the proposed action, the records used, the unresolved issue and any action already taken. It should be possible to decide what happens next without repeating the whole investigation.

This creates a management obligation. Someone must own the queue, have authority to resolve it and be available within a stated service window. If no one responds, the system needs a defined waiting state. Silence must not quietly become approval.

The NIST Generative AI Profile treats risk as dependent on the system and its use, within a voluntary risk-management framework. That supports examining the actual workflow rather than assigning an agent a single, reassuring safety label. It does not supply an organisation's escalation thresholds or guarantee a safe deployment.

Test the awkward cases before widening access

An evaluation built entirely from ordinary requests can conceal the cases that make authority expensive. The pilot should include missing orders, duplicate requests, changed account details, unavailable services and contradictory policy documents. Each case needs an expected response before the test begins.

Recovery deserves particular attention. If a payment request times out, repeating it immediately may be inappropriate: the first request could already have succeeded. The system should reconcile the transaction through an authoritative record before deciding to retry. A customer-facing message should distinguish a pending confirmation from a completed refund.

The review should measure more than the proportion of tasks completed. Useful proposed measures include accepted outcomes requiring no correction, exceptions per task, time spent reviewing exceptions and time needed to recover from a failed action. These are evaluation recommendations, not results from a deployment described here.

Reviewers should examine the cases behind an aggregate score. A system that handles routine questions well but mishandles account changes may still be useful within a narrower permission boundary. Expanding access should follow evidence about the newly permitted action, rather than confidence borrowed from unrelated successes.

Avoid turning every action into an approval queue

There is a fair objection: requiring human approval everywhere can remove much of the value of automation. A customer gains little if a supposedly autonomous assistant simply creates another queue.

The answer is to vary authority with consequence and reversibility. Drafting a response, retrieving a record and making a financial commitment deserve different controls. A business can permit routine, bounded actions while reserving unusual or consequential cases for a person. The boundary should be explicit enough to implement and revise.

Nor should a good pilot make that boundary permanent. Policy changes, new tools and altered customer behaviour can invalidate earlier assumptions. The owner needs a way to reduce access, investigate incidents and retest the affected paths. Expanding autonomy should be a reversible operational decision where possible.

The approval worth seeking is therefore specific: this system may perform these actions, using these records, under these conditions. These exceptions go to this owner. These failures suspend the permission.

That is a more demanding brief than asking whether an agent can complete a task. It also gives the business something useful to govern after the demonstration ends.

Sources

Anthropic, Building effective agents.

NIST, Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, introduction and risk overview.