XenGrowth

AI WORKFLOW GOVERNANCE

The AI Agent Governance Checklist: Permissions, Review, and Recovery Before Launch

Before an AI agent touches a CRM, an inbox, or an ad account, it needs scoped permissions, an approval gate, and a way to shut it off. Here is how to build all three.

An implementation guide · Reviewed September 7, 2026 · 9 min read
xengrowth.com/resources/ai-agent-governance-checklist

Contents

01

What actually needs this level of control, and what does not

Not every automated workflow needs a permissions matrix, an approval gate, and a kill switch. A rule-based automation — a form submission that triggers a fixed email sequence, a webhook that copies a new lead into a CRM field, a scheduled report that runs the same query every Monday — follows a predetermined path. Someone reviewed that path once, at build time, and it does not change unless a person edits it. The risk surface is small and visible from the outside: check the trigger, check the action, done.

An agent is different. For this checklist, an agent is a system that takes multiple steps toward a goal, decides between those steps on its own, and calls tools or APIs without a person confirming each call. It might pull a contact's history, decide which of several follow-up sequences fits, draft the message, and send it — choosing both the sequence and the wording itself rather than filling in a fixed template. That decision-making step is exactly where governance has to apply, because nobody wrote down in advance which specific action the system would take in that instance.

- It chains two or more tool or API calls in a single task without a human confirming each one.
- The sequence of actions is not fixed in advance — the system decides which branch to take based on the data it sees.
- It can take an action a person did not review in that specific instance (send, update, publish, bid) rather than only producing a draft or a recommendation.
- Its behavior on the same input can change over time — a model update, a prompt edit, different retrieved context — without a corresponding code change to review.

> *That decision-making step is exactly where governance has to apply, because nobody wrote down in advance which specific action the system would take in that instance.*

02

What should this specific agent be allowed to touch

OWASP's Top 10 for LLM Applications names excessive agency as a top-tier risk for a reason: agent incidents are usually a scope problem, not a reasoning failure. OWASP breaks it into three failure modes — excessive functionality (the agent has access to tools it does not need for its actual task), excessive permissions (a tool it does need can do more than the task requires, like a mailbox connector that can also delete messages), and excessive autonomy (a high-impact action executes with no human check). All three get fixed the same way: write down, per task, exactly which systems the agent can reach, exactly what it can do in each one, and which of those actions still require a person's sign-off before they execute.

The matrix below is a starting shape, not a template to copy verbatim — the specific systems, bands, and thresholds should match what each workflow actually needs and what a mistake there would actually cost.

System / Tool	Action Allowed	Requires Approval?	Notes
CRM records	Read contact and deal history	No	Scope to records relevant to the active task, not the full database.
CRM records	Update lead status or add a note	No	Reversible and logged; treat as low-risk.
CRM records	Delete or merge records	Yes	Usually irreversible without a full backup restore.
Email / outreach	Draft an email to a known contact	No	Draft only — see send permission below.
Email / outreach	Send an email to an external contact	Yes, until a track record is established	Move to sampled review only after a defined accuracy history.
Calendar	Book or cancel a meeting on someone else’s calendar	Yes	Affects another person’s time directly.
Ad platforms	Adjust bids within a preset band	No	Cap the band tightly enough that a bad decision is cheap.
Ad platforms	Raise daily budget beyond the band, or launch a new campaign	Yes	Budget changes are effectively spending decisions.
Billing / payments	Any write action — refund, charge, plan change	Yes, always	No unsupervised exception; see approval gates below.
CMS / website	Publish to the live site	Yes	Public-facing and hard to fully unwind once indexed or seen.

03

Which actions actually need a person in the loop, and why

An action needs a person's sign-off first when getting it wrong is expensive to undo, not merely embarrassing to explain. Three categories consistently meet that bar: irreversible actions, anything that moves money, and anything a customer or prospect sees directly. Reversible mistakes — a mistagged lead, a note added to the wrong field, a draft nobody sent — cost a few minutes to fix. Irreversible ones do not have that safety margin: deleting a record with no backup, merging two customer accounts, canceling a subscription, or sending an email that cannot be unsent once it lands in an inbox.

Anything touching money deserves its own gate even when it is technically reversible, because financial mistakes carry trust and compliance costs beyond the dollar amount involved. A refund issued to the wrong account, a subscription downgraded instead of upgraded, an ad budget raised tenfold by a misread instruction — these are the actions most agent-governance guidance singles out first. A person can catch a financial error in seconds that would otherwise take a support team days to trace and unwind.

Customer-facing actions need a gate for a different reason: they represent the business to someone who has no way to tell whether a person or a system produced them. An agent that drafts a support reply is low risk; one that sends that reply unreviewed is making a representation in the company's voice with no second reader. The same logic applies to a social post, a price quote, or a contract clause — anywhere a mistake lands directly in front of a customer before anyone internal sees it.

04

What has to be logged for every action an agent takes

An audit trail that only records the agent's final output is not an audit trail — it is a transcript. For governance purposes, every action needs four things captured together: what the agent did (the specific tool call or write, not a paraphrase), why it did it (the input, retrieved context, or reasoning summary that led to that action), what data it touched (which records, files, or systems), and what changed as a result. Without all four, someone investigating a bad outcome has to reconstruct intent from a result after the fact, which is slow and often impossible once the underlying data has already changed again.

Retention should match how long a dispute or a compliance question could realistically surface, not how long storage happens to be cheap. For most B2B contexts, that means keeping agent action logs at least as long as the business already retains related records — CRM audit history, financial records, support tickets — rather than inventing a shorter, agent-specific window. NIST's AI Risk Management Framework treats traceability and documentation as a baseline governance function, on the reasoning that a system nobody can audit after the fact is a system nobody can actually be held accountable for.

- Timestamp, workflow identifier, and the specific model or prompt version in use at the time
- The tool or API call made, including its parameters, not a summary of the action

- The record, account, or file touched, by identifier
- The before-state and after-state of anything modified, or a diff
- Whether the action passed through a human approval gate, and who approved it

> *Retention should match how long a dispute or a compliance question could realistically surface, not how long storage happens to be cheap.*

05

What a kill switch and a rollback actually need to do

A kill switch is not a single button; it is three separate capabilities, built and tested before launch rather than improvised during an incident. First, stopping new tasks: the agent should be revocable at the credential or permission level, not just by disabling a workflow trigger that a retry queue might bypass. Second, freezing in-flight work: a multi-step task interrupted mid-sequence should not be left in an undefined state where half its actions completed and half did not. Third, knowing what already happened: without the audit trail described above, there is no reliable way to know which actions to unwind once the agent itself has been stopped.

Recovery means something more specific than turning the system off. It means tracing every logged action back to the last point the system is known to have behaved correctly, reverting the data changes that can be reverted, and separately listing the ones that cannot — a sent email, a public post, a notified customer — because those need a human response rather than a technical rollback. It also means telling anyone affected before they discover the problem themselves, which is a communications step as much as a technical one. The joint CISA/NSA guidance on deploying AI systems treats incident response planning as inseparable from the deployment decision itself: a system without a tested recovery process is not ready for production, regardless of how well it performs on ordinary tasks.

06

What to ask a vendor before their agent gets production access

A third-party agent platform inherits every governance requirement above, plus one more: the company operating it, not just the model underneath it, decides how much of this checklist is even possible for a customer to implement. Before granting a vendor's agent write access to a live system, get specific answers in writing rather than a sales deck's general reassurance.

- Can we export a complete action log — every tool call, not a summarized activity feed — in a format our own systems can read?

- What is the default permission scope, and can we narrow it per workflow, or is it all-or-nothing for the whole integration?
- Is human approval for irreversible, financial, or customer-facing actions built into the product, or is that left to us to build on top of it?
- How quickly can we revoke access — is it an action we control directly, or a support ticket with a queue?
- What happens to our data, and to logs of the agent's past actions, if we cancel or downgrade the plan?
- Has the vendor assessed its own product against a recognized framework, such as the OWASP Top 10 for LLM Applications or NIST's AI Risk Management Framework, and will they share that assessment?

> *Before granting a vendor's agent write access to a live system, get specific answers in writing rather than a sales deck's general reassurance.*

07

What this checklist does not fix

None of this eliminates the hardest failure mode in agent governance: an output that is wrong in a way that looks completely normal. A permissions matrix stops an agent from deleting records it should not touch. An approval gate stops it from sending something before a person sees it. Neither one stops a human reviewer from approving a plausible-sounding email that cites the wrong contract term, or a summary that quietly drops a caveat that changes its meaning — because the reviewer is checking whether the output looks right, not independently re-deriving whether it is right. That gap tracks with a well-documented pattern in automation more broadly: people calibrate trust in a system based on how it has performed recently, and a system that is usually right tends to get less scrutiny over time, not more.

The practice that compensates is not a better prompt or a stricter gate. It is a standing habit of sampling: periodically — weekly for a new workflow, monthly once it has stabilized — someone redoes a small number of the agent's completed tasks from scratch, independently, and compares their own answer to what the agent produced and a person already approved. That catches the one failure mode approval gates cannot, structurally: not the action that should have been blocked, but the one that looked exactly like it should have been approved.

Key Takeaways

- Before granting any permission, write down the specific system, the specific action, and whether it is reversible — read access and delete access are different permissions, not one line item.

- Require a human approval step for any agent action that is irreversible, moves money, or reaches a customer directly, even after the agent has handled similar actions correctly for months.
- Log the input, the exact tool call, the data touched, and the resulting change for every agent action, not just the final output, and retain those logs as long as related business records are kept.
- Build the kill switch before launch: know exactly who can revoke an agent's credentials, how long that takes end to end, and what happens to work already in progress when it fires.
- Ask a third-party agent vendor to show you their action logs and default permission scope in writing before granting production access, not after an incident forces the question.

Sources

1. AI Risk Management Framework — NIST — https://www.nist.gov/itl/ai-risk-management-framework
2. NIST AI 600-1: Artificial Intelligence Risk Management Framework — Generative Artificial Intelligence Profile — https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf
3. OWASP Top 10 for Large Language Model Applications — https://owasp.org/www-project-top-10-for-large-language-model-applications/
4. LLM06:2025 Excessive Agency — OWASP Gen AI Security Project — https://genai.owasp.org/llmrisk/llm062025-excessive-agency/
5. Agentic AI — Threats and Mitigations — OWASP Gen AI Security Project — https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/
6. Joint Guidance on Deploying AI Systems Securely — CISA — https://www.cisa.gov/news-events/alerts/2024/04/15/joint-guidance-deploying-ai-systems-securely