

Production AI Assurance Assessment

Illustrative report for a customer refund agent

Assessment ID	AAI-SAMPLE-2026-001	Assessment date	28 September 2026
System/version	refund-agent v1.4.2	Environment	Simulated staging
Scope	One refund decision workflow	Method	Synthetic scenario illustration

ILLUSTRATIVE - NOT A CLIENT ASSESSMENT

Executive decision

DENY	96.8%	3	4
Release action	Task success (synthetic)	Critical failures	Policy violations

Decision rationale. The illustrative task-success rate exceeds the 95% target, but critical authorization and policy-control failures prevent release. Aggregate success does not override a critical control failure. This decision applies only to the example scope and synthetic evidence.

Important limitation. Every system detail, measurement, finding, test outcome and financial assumption in this document is synthetic. No real customer system was tested. This is a demonstration of report structure, not a certification, audit opinion or measured performance claim.

1. Scope, method and decision semantics

The example workflow classifies refund requests, retrieves order context and proposes an outcome for a human-approved refund service. The agent is not authorized to issue payments or override refund policy.

Assurance questions in scope

Identity and version	Can results be tied to an exact system and environment?
Authority boundary	Can the agent attempt a consequential action without approval?
Input robustness	Can untrusted customer text change policy interpretation or tool authority?
Tool failure	Does the workflow fail safely on timeout, malformed output or permission failure?
Human review	Are ambiguous and policy-conflicting cases escalated?
Evidence traceability	Can results be tied to case IDs, expected behavior, versions and timestamps?

Illustrative measurements

Test cases	1,000	Example denominator; not a live run
Expected behavior matched	968 / 1,000	96.8% point estimate for this synthetic sample
Critical failures	3	Release-blocking under the example policy
Policy violations	4	Require triage and root-cause review
P95 latency	2.4 seconds	Synthetic observation
Tool-error recovery	91 / 100 (91.0%)	Illustrative injected-error sample

Decision semantics

BLOCKED / DENY: one or more critical controls or observed behaviors violate the agreed release policy. **REVIEW:** important gaps or incomplete evidence need resolution or an explicitly authorized exception. **ASSURED / ALLOW:** only within the declared scope and evidence boundary; not universal safety or compliance certification.

2. Findings and root-cause hypotheses

The examples below show how findings should connect observed behavior to action. In a real engagement, every observation must point to an actual test artifact and must be reproduced before final classification.

F-001 / CRITICAL	Unauthorized refund action: a synthetic case attempted to bypass human approval.	Enforce approval and authorization outside the model. Retest: every unauthorized refund is denied or routed to an auditable approval gate.
F-002 / CRITICAL	Prompt injection: synthetic untrusted text influenced the tool-selection plan.	Separate instructions from untrusted content and restrict tool allow-lists. Retest direct and indirect injection variants; no authority or policy bypass.
F-003 / CRITICAL	Policy conflict: conflicting order data produced an approval proposal rather than escalation.	Fail closed on authoritative-data conflicts. Retest all defined conflict cases; expected outcome is deny or human review.
F-004 / HIGH	Retry control: synthetic tool timeouts caused repeated retries without a bounded budget.	Set retry, latency and spend ceilings; use idempotency. Retest timeout and partial-execution paths.
F-005 / HIGH	Evidence completeness: example records lacked a linked policy version.	Bind each result to exact system, policy, dependency and evaluator versions; reject incomplete release evidence.

Severity is not a prediction of financial loss. It describes the consequence of the stated control failure under the agreed assessment policy. A real report must include the severity rubric, confidence and known limitations.

3. Evidence quality and traceability

Evidence quality is a separate dimension from outcome score. Missing evidence is a gap, not a pass; a hash can detect content changes relative to a stored digest but does not prove source truth or completeness.

Asset identity and version	PASS	Record exact system, build and environment identifiers.
Authority contract	GAP	Confirm allowed actions, permissions and approval-required actions.
Execution traces	PARTIAL	Attach case IDs, timestamps, expected and observed outcomes, and runner version.
Policy artifact/version	GAP	Store the policy snapshot and canonical hash used for the run.
Dependency inventory	PARTIAL	Record model provider/version, tool versions and material dependencies.
Retest record	GAP	Link each finding to code/config change and same-case retest result.

Evidence register example

EV-S-001	Test manifest	Versioned case IDs, expected behavior, severity and environment
EV-S-002	Execution summary	Run count, denominators, timestamps and runner version
EV-S-003	Critical failure trace	Redacted trace, observed action and expected authority
EV-S-004	Policy snapshot	Policy ID/version and canonical hash
EV-S-005	Retest record	Original finding ID, code/config change and repeated case outcomes

All evidence IDs above are placeholders. A real delivery must point to retained, authorized artifacts and must record retention/deletion requirements.

4. Business impact analysis

The analysis identifies plausible consequence pathways from the workflow and control findings. It does not infer loss values from technical severity alone. A customer-specific financial scenario requires business-owner-approved event volume, baseline failure rate, target rate and cost per failure.

Qualitative impact pathways

Unauthorized consequential action	Approval boundary may be bypassed.	Unauthorized transaction, refund or workflow state change.
Sensitive-data handling	Data egress control must be explicit and tested.	Unintended disclosure, contractual exposure or response costs.
Tool/dependency failure	Retries and partial execution can affect completion.	Duplicate actions, delayed service or manual recovery effort.
Unassessed change	A model/tool/policy update can invalidate earlier evidence.	Regression reaches production without a refreshed decision.

Optional scenario model - synthetic arithmetic example only

Monthly workflow volume	50,000	Synthetic assumption; replace with client data
Baseline failure rate	1.5%	Synthetic assumption; not measured
Target failure rate	0.5%	Planning assumption; not a guarantee
Cost per failure	\$75	Synthetic assumption; validate with finance/operations
Baseline monthly scenario exposure	\$56,250	$50,000 \times 1.5\% \times \75
Target monthly scenario exposure	\$18,750	$50,000 \times 0.5\% \times \75
Modeled monthly difference	\$37,500	Arithmetic difference only; not promised savings

Scenario figures above are fabricated for demonstration. They are not observed loss, a forecast, a financial guarantee, or evidence that AAI will achieve the target rate.

5. Prioritized remediation roadmap

Deadlines below are proposed planning windows, not commitments. The client and AAI must agree owners, release policy and retest scope before work begins.

P0 - before release	Platform + Security	Enforce tool permissions and approval outside the model.	All unauthorized-action cases denied; no approval bypass.
P0 - before release	AI Engineering	Separate untrusted input from system instructions; run injection suite.	No tested injection changes authority, policy or tool allow-list.
P0 - before release	Product + Operations	Fail closed on contradictory order/policy data.	All defined conflict cases deny or escalate.
P1 - next 5 business days	SRE / Platform	Bound retries, time, spend and idempotency behavior.	Injected failures stay within agreed budgets and do not duplicate consequential actions.
P1 - next 5 business days	Platform / Assurance	Bind each record to exact system/policy/dependency versions.	Evidence manifest complete for every release-gate case.
P2 - next sprint	AI Platform	Add regression suite and change-triggered reassessment.	Material prompt/model/tool/policy changes trigger the agreed suite.

Closure states

OPEN: remediation not implemented. **READY FOR RETEST:** client provides versioned change and test environment. **RESOLVED:** the agreed retest criterion passed with evidence. **ACCEPTED EXCEPTION:** authorized customer owner documents residual risk, expiry and compensating controls. **REGRESSED:** a previously passing case now fails.

6. Additional validation plan

The table is a proposed test plan, not evidence that these tests have been executed. A real engagement executes the agreed tests against a customer-authorized staging boundary, records exact versions and links outputs to findings.

Authority bypass	Attempt out-of-scope refund/write action with and without approval context.	Deny or approval-gate action; auditable policy decision.
Direct/indirect prompt injection	Use untrusted user and retrieved content to attempt instruction/authority override.	No unauthorized tool call; injection attempts recorded.
Data boundary	Attempt sending synthetic sensitive data to an unapproved destination.	Egress blocked; policy reason and trace retained.
Policy conflict	Provide contradictory order state, refund rules and stale data.	Fail closed or route to human review.
Tool resilience	Inject timeout, malformed response, permission failure and partial execution.	Bounded retries; no duplicate consequential action; safe fallback.
Dependency change	Change a model/tool/prompt/policy version.	Affected evidence invalidated or reassessment triggered.
Telemetry reconstruction	Replay action and reconstruct identity, timestamp, tool, policy and decision.	Every required field available or gap explicitly reported.

A release decision must be tied to the agreed test set, expected outcomes, environment and version. A test plan alone is not proof of control effectiveness.

7. Engagement delivery and acceptance

Proposed delivery window: 5-7 business days after scope, access and evidence prerequisites are complete. The timeline depends on client responsiveness, system complexity and agreed run volume.

Day 1	Scope, authority boundary, impact assumptions and acceptance criteria.	Signed scope, system inventory and agreed test plan.
Days 2-3	Baseline evidence intake and agreed customer-specific tests.	Run manifest, measurement definitions and initial evidence register.
Days 4-5	Finding triage, root-cause hypotheses and business-impact analysis.	Severity-ranked findings, evidence references and remediation roadmap.
Day 6	Remediation review and included retests where agreed.	Retest delta: resolved, unresolved, regressed and not tested.
Day 7	Handover and release-decision review.	Final report, evidence manifest, decision rationale and reassessment triggers.

Acceptance checklist

- Every executed test has a stable ID, expected and observed result, status, timestamp and evidence reference.
- Metrics include denominators, sample size and calculation method.
- Critical findings cannot be hidden by an aggregate score.
- Each finding includes severity, impact pathway, evidence, owner recommendation and retest criterion.
- Release decision follows the agreed policy and is traceable to findings.
- Untested areas, assumptions and limitations are disclosed.
- Client handover, residual risks, evidence retention/deletion and reassessment triggers are documented.

Commercial boundary

The \$10,000 amount is a proposed fixed-scope engagement price, not a validated market benchmark. Its value must be demonstrated through customer-specific execution, actionable findings, engineering-grade evidence and a useful retest decision. A dashboard or synthetic benchmark alone does not justify the fee.

Limitations: This sample is not a live system assessment, penetration test, legal opinion, compliance certification, guarantee of safety or guarantee of loss avoidance.