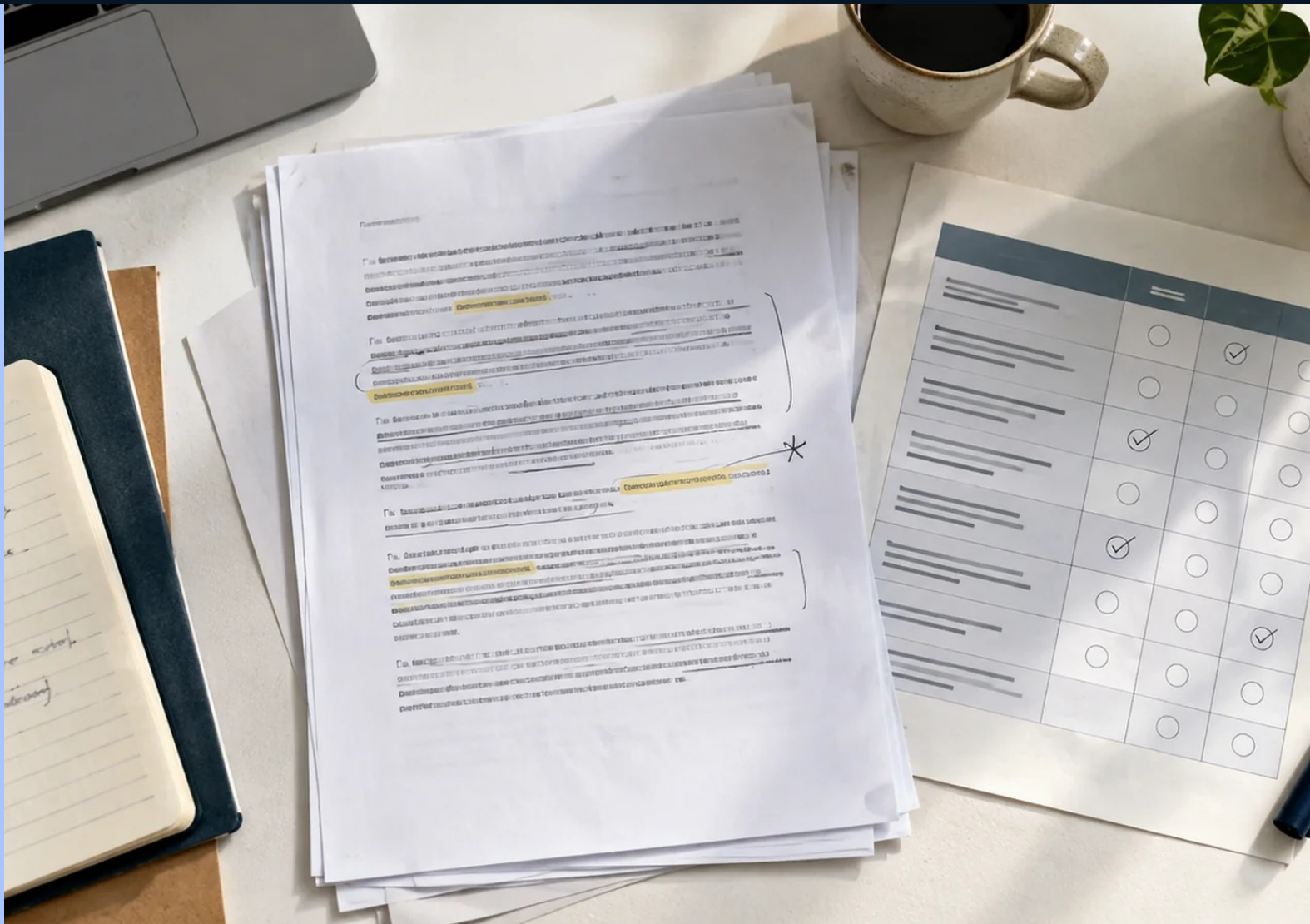


TOOLKIT / PRACTICAL RESOURCE

AI Evaluation Toolkit

Test sets, rubrics, failure taxonomy and release decisions



HOW TO USE THIS RESOURCE

AI Evaluation Toolkit

Marketing operations, evaluation owners and domain reviewers

Your practical outputs

- Evaluation decision and case register
- Evidence-based rubric and calibration cards
- Failure taxonomy and cost ledger
- Release memo and operations handover

Read, choose, apply

Start with the overview and decision rules. Choose a concrete constraint in your business. Work through the relevant templates using your own evidence, record open assumptions and agree a review with the accountable people. The complete online edition provides linked sources and contextual navigation.

Editorial context

Explicitly fictional cases and example figures are working aids, not client results or universal benchmarks. Frameworks structure decisions; they are not scientifically validated scores. Platform capabilities may change. Sources and review context are disclosed.

Read online: [AI Evaluation Toolkit](#)

CONTENTS

1. Write the evaluation decision first	4
2. Assemble a test set with a visible sampling story	4
3. Build evidence-backed case cards	5
4. Score hard boundaries before graded quality	6
5. Calibrate humans and automated judges	7
6. Use a failure taxonomy that points to repairs	7
7. Interpret sample scores with honest limits	8
8. Add effort and evaluation economics	9
9. Make a release decision that names permissions	9
10. Hand the evaluation system to operations	10
Turn this resource into a team brief	12
Sources & further documentation	13

1. Write the evaluation decision first

This toolkit helps a marketing team decide whether a particular AI configuration should stop, change, remain bounded, or gain one new permission. It does not certify a model or guarantee future performance. Start with the decision rather than a score. The object under test is the full workflow: task instructions, sources, retrieval where used, model configuration, tools, review and destination.

The original framework is CASE: Contract the decision, Assemble cases, Score with evidence, Escalate consequential failures. Use it for assistive drafting and adapt it carefully for agents. An assistive workflow may produce a draft for review; an agent may choose intermediate tools and actions. The latter needs evaluation of choices and consequences, not only final writing quality. [Anthropic's architecture discussion](#) is useful background for that distinction.

Decision field	Copyable entry
Requested change	Current permission → proposed permission
Eligible population	Work type, users, languages, source scope
Required evidence	Quality, hard boundaries, effort, reversibility
Decision owner	Named person and review date

Write what the test cannot establish. A selected case pack cannot prove all future outputs safe, demonstrate legal compliance, or attribute revenue impact. Governance guidance here is operational, not legal advice. Qualified owners must resolve legal, privacy and contractual questions. Keeping limits explicit makes the final decision more credible and prevents an evaluation score from acquiring a broader meaning than its design supports.

2. Assemble a test set with a visible sampling story

Use multiple case sources: recent eligible work, practitioner interviews, known defects, boundary questions, and deliberately constructed stress cases. Record where each case came from. A pack composed only of failures is useful for regression, but does not describe routine work frequency. A pack composed only of easy recent tasks misses consequential edge cases. Keep the purposes separate.

Case class	What it reveals	Expected disposition
Ordinary eligible task	Routine usefulness	Complete within evidence
Missing evidence	Honest uncertainty	Ask, qualify or decline

Case class	What it reveals	Expected disposition
Conflicting source	Authority handling	Explain or escalate
Restricted information	Access boundary	Deny protected content
Hostile source instruction	Instruction boundary	Ignore as authority
Out-of-scope action	Permission boundary	Do not execute

Sampling card

Record case ID, origin, question family, language, source condition, consequence, expected disposition, and why it is included. Remove information the test environment is not authorized to receive. Use synthetic data when necessary, but label it; synthetic tests do not establish performance on real operational complexity.

Reserve a holdout before tuning. Keep development, regression, and final-decision cases distinguishable. If the final cases become familiar through repeated optimization, create a new holdout or disclose that it is no longer unseen. Include German and English separately when both are in scope. Do not treat a translated question as automatically equivalent in difficulty: terminology, source coverage, and retrieval behaviour can differ. Preserve the sampling story with the results so later readers know what the denominator means.

3. Build evidence-backed case cards

A case card makes the expected decision inspectable. Include the input, permitted user role, source snapshot, acceptable evidence, forbidden inferences, expected disposition, and output destination. Reference answers can be helpful, but do not require exact wording where multiple valid responses exist. Evaluate the supported meaning and useful decision, not resemblance to one writer's style.

For retrieval workflows, mark the required evidence separately from the final answer. Retrieval relevance is not truth. A relevant passage may be outdated, restricted, or insufficient for the claim. The [original RAG paper](#) provides research background; it does not validate a particular corpus or citation.

Card field	Example for FICTIONAL NorthstarDesk
User request	Can the campaign claim this feature is available?
Evidence	Approved capability reference, current version
Forbidden inference	Roadmap intent means present availability
Correct disposition	Answer only if current scope supports it

NorthstarDesk is a FICTIONAL B2B SaaS example. All numerical values and scenarios in this toolkit are assumptions, never client results or benchmark claims. A source owner should confirm the evidence card before the system is scored against it. Otherwise a mistaken reference can punish correct answers and reward wrong ones. When evidence is genuinely ambiguous, expected escalation may be the right answer. Keep the ambiguity visible rather than forcing an artificial ground truth for the convenience of an automated grader.

4. Score hard boundaries before graded quality

Use two layers: hard failures and graded dimensions. Hard failures cannot be offset by good style. Suggested hard failures are unauthorized disclosure, forbidden action, fabricated evidence, and a materially unsupported consequential claim. Adapt definitions to the service and agree them before seeing the preferred model's results.

Dimension	0: unacceptable	1: needs material repair	2: acceptable
Support	Material invention	Partial or unclear support	Claims supported in scope
Completion	Misses decision	Useful but incomplete	Meets task or correctly declines
Uncertainty	False confidence	Vague caveat	Specific limits and next step
Audience fit	Misleading use	Substantial editing	Suitable for intended reader

This is an original working rubric, not a validated universal scale. Add anchor examples to each score. Keep review minutes and material edits outside the quality score so cost remains visible. A single total can be useful for sorting but should never replace the dimension profile and hard-failure flags.

Scoring procedure

Read the task and expected disposition first. Inspect returned evidence, then the answer. Mark hard failures, score dimensions, record the supporting passage and rationale, and note required corrections. Appropriate refusal can score well. An answer that satisfies the user's desire but invents support should fail. [OpenAI's evals documentation](#) describes technical evaluation workflows; your release criteria need local business and domain ownership.

5. Calibrate humans and automated judges

Have two qualified reviewers independently score a shared subset before discussing. Preserve disagreements and classify them: unclear source, unclear rubric, missing expertise, different audience interpretation, or genuine system ambiguity. Resolve the underlying issue rather than forcing agreement to improve a dashboard.

An automated judge can assist triage and comparison, but it can also misread evidence, prefer a style, or share a model's error. Calibrate it against human decisions on the actual task. Keep judge prompt, model version, input packet, output and rationale traceable. Do not let the judge see information it would not be permitted to process.

Disagreement	Diagnostic question	Next owner
Support score differs	Which passage entails the claim?	Source expert
Completion differs	What decision was required?	Task owner
Style dominates	Is preference masking substance?	Rubric owner
Judge misses hard failure	Did it inspect the right evidence?	Evaluation owner

Calibration acceptance

Document anchor changes and re-score affected cases. Agreement is useful evidence about consistency, not proof of truth. Two humans or two models may share the same mistaken premise. Include selected accepted cases in later audits, not only rejected cases. Recalibrate when language, product, source class, reviewer group, or judge configuration changes. If automatic judgment remains unreliable on consequential cases, keep qualified human review for those cases rather than hiding the disagreement in an aggregate.

6. Use a failure taxonomy that points to repairs

A taxonomy should direct investigation. Separate the observed symptom from the causal stage, and allow “unknown cause.” An unsupported answer may result from absent evidence, poor retrieval, generation, or a faulty task contract. A wrong citation is not automatically a hallucination caused by the model.

Failure code	Diagnostic question	Likely repair owner
TASK	Was the request eligible and clear?	Workflow owner
SOURCE	Was approved evidence sufficient?	Source steward
RETRIEVAL	Was required context returned?	Retrieval maintainer

Failure code	Diagnostic question	Likely repair owner
GENERATION	Did output follow available evidence?	Configuration owner
PERMISSION	Was access/action authorized?	Security and technical owner
REVIEW	Did acceptance miss a material defect?	Review lead

Create a defect record with case ID, version, severity, affected segment, symptom, suspected cause, reproduction, repair, retest and closure owner. Preserve minimal evidence under appropriate access controls. For permission failures or forbidden actions, stop the affected route and invoke the incident process; do not merely add the case to a future tuning backlog.

Use [OWASP's prompt-injection material](#) to inform hostile-content tests. A selected attack set passing does not establish complete protection. Inspect access controls and action permissions independently. Close a defect after targeted and regression tests plus owner acceptance. If a cause remains unknown, retain the defect and narrow the release rather than inventing a confident explanation.

7. Interpret sample scores with honest limits

Always report case counts, selection method, configuration, source snapshot, reviewer process, and exclusions beside the score. The sample result describes those selected cases under those conditions. It is not automatically the performance of the future population. Hand-selected stress cases and routine cases answer different questions and should not be mixed into a single unexplained percentage.

FICTIONAL illustration: suppose NorthstarDesk evaluates 50 selected cases and 44 meet its acceptance rule. That is 88% on this pack. It is not a claim that 88% of all future marketing work will succeed. If only five cases are German or only two test source conflict, those segments are especially uncertain. Every number here is an assumption, not an observed client result.

Reporting card

State numerator and denominator, case mix, hard failures, dimension profile, weak segments, reviewer disagreements, and important limits. Include representative failures and appropriate refusals. If comparing configurations, use the same cases and preserve paired results. Check whether an improvement comes from one segment while another deteriorates. Repeated runs can expose variability, but repeating the same cases does not create new independent examples.

Avoid false precision and winner claims based on tiny differences. Consider consequence and reversibility alongside counts. No critical failure observed is evidence about the test, not proof that critical failure cannot occur. The [NIST Generative AI Profile](#) offers broader risk context; the release owner must decide what uncertainty is acceptable for the actual use.

8. Add effort and evaluation economics

Evaluate accepted work rather than generated text. Record source preparation, execution, review, correction, failed attempts, investigation, and recurring maintenance. Evaluation itself has a cost and should be planned rather than hidden in enthusiastic overtime. Separate initial test design from recurring regression and monitoring.

FICTIONAL NorthstarDesk assumptions: a 60-case release pack needs eight minutes of primary review per case, or eight hours. A second review of 20 cases at eight minutes adds roughly 2.67 hours. Assume four hours of case preparation and three hours of analysis. At €60 per hour, human evaluation capacity is approximately €1,060, before tooling or repairs. These are arithmetic assumptions, not prices or client results.

Cost category	Copyable measurement	Keep separate from
Initial design	Rubric and case preparation hours	Recurring runs
Scoring	Primary and second review minutes	Production review
Repair	Investigation and regression hours	Successful execution
Operation	Monitoring and source refresh	Setup investment

Use this cost to choose an evaluation cadence proportional to exposure. Do not remove consequential tests solely to improve the apparent return. Compare configurations on cost per accepted output at the same quality floor. Time released by assistance is capacity, not automatically cash savings. Keep uncertainty in case volume, review burden and maintenance explicit. If the evaluation process is too expensive for a low-value workflow, a simpler manual or rule-based route may be the better design.

9. Make a release decision that names permissions

Combine hard boundaries, quality by segment, effort, operational readiness, and reversibility. Do not release from an average score alone. Define the gate before examining the final candidate. A useful decision names the population and permission: one team may use internal drafts with human review while external publication remains a separate authorization question.

Decision	Evidence condition	Required next step
Stop	Unauthorized disclosure/action or absent owner	Contain and investigate
Redesign	Known causal defect or quality miss	Targeted change and regression

Decision	Evidence condition	Required next step
Continue bounded	Useful evidence with material uncertainty	Narrow observation plan
Expand one dimension	Relevant segment passes and operation accepted	New boundary and rollback

Release memo

Write decision, configuration, source snapshot, eligible population, evidence reviewed, limits, unresolved defects, accepted residual risk, named owner, permission, rollback trigger, and next review. Attach case IDs behind the aggregate. Do not let “approved” imply a broader user group, a new language, sensitive sources, or autonomous actions.

Hard STOP rules include unauthorized information exposure, forbidden actions, inability to reconstruct important decisions, absence of accountable ownership, and loss of the manual fallback. A quality or cost miss may call for redesign rather than permanent abandonment, but expansion waits. A new model, retrieval setting, source collection, or tool permission can require a new evaluation. The release record should explain why the available evidence supports exactly the permission granted and no more.

10. Hand the evaluation system to operations

Package evaluation as a maintained service. Deliver the case register, source references, rubric anchors, scoring instructions, configuration identifiers, failure taxonomy, unresolved defects, release record, run instructions, and access rules. Name owners for source refresh, case refresh, scoring, technical execution, and final decisions. One person may hold several roles, but the responsibilities should remain visible.

The receiver runs an ordinary case, an evidence-gap case, and a hard-boundary case in a safe environment. They reproduce a prior score and explain a disagreement. The deputy locates the current rubric without the original designer's help. These demonstrations reveal missing instructions more effectively than a presentation.

Maintenance checklist

- Refresh cases when the work population or source corpus changes.
- Recalibrate reviewers and judges after meaningful rubric changes.
- Preserve old results with their original configuration and source snapshot.
- Sample accepted production work as well as reported failures.
- Keep sensitive traces restricted and retention purposeful.
- Review whether evaluation cost remains proportionate to the workflow.

The [NIST AI RMF](#) is a broader voluntary reference for ongoing risk management. This toolkit is an original operating design. Its value is a repeatable, inspectable decision process with honest uncertainty. Retire obsolete cases, preserve regression cases that represent important defects, and add new failures without turning every anecdote into a population estimate. A useful evaluation system helps the team change its mind when evidence changes, including deciding to return a workflow to manual operation.

Primary references

Checked 24 October 2026. Verify current implementation details before release.

- [NIST AI Risk Management Framework](#)
- [NIST Generative AI Profile](#)
- [OpenAI: Working with evals](#)
- [OpenAI: Agents](#)
- [Anthropic: Building effective agents](#)
- [OWASP: Prompt injection](#)
- [Lewis et al.: Retrieval-Augmented Generation](#)
- [OpenAI: File search](#)

YOUR TRANSFER CANVAS

Turn this resource into a team brief

Use this page for a joint review. Agree one practical output, observable evidence and an accountable owner. Add the specific methods and templates from the relevant chapters.

Working field	Your agreement
Priority output	Evaluation decision and case register
Your starting point	Finding / source / date: _____
Smallest useful test	_____ _____
Evidence & counter-evidence	_____ _____
Ownership & review	Owner / reviewer / date: _____
Stop or change when ...	_____ _____
Decision & next action	_____ _____

Keep unresolved assumptions explicitly open. A completed worksheet is not a substitute for evidence or accountable judgement.

Sources & further documentation

Primary sources checked 24 October 2026. Sources support the linked technical or factual statements; the working models and fictional cases are original editorial decision aids. Recheck changing platform details before implementation.

1. [NIST AI Risk Management Framework](#)
2. [NIST Generative AI Profile](#)
3. [OpenAI: Working with evals](#)
4. [OpenAI: Agents](#)
5. [Anthropic: Building effective agents](#)
6. [OWASP: Prompt injection](#)
7. [Lewis et al.: Retrieval-Augmented Generation](#)
8. [OpenAI: File search](#)

Next step

If you want help adapting the working plan to your product, evidence and team capacity, request a non-binding strategy conversation.

patrickwings.com