



AI evaluation and quality assurance

Measuring whether AI outputs and workflows are dependable enough for their purpose.

WORKS WITH THE ASSISTANTS TEAMS ALREADY USE



EDITION 15 / EMBER & VIOLET

EXECUTIVE WORKING EDITION - 08 OCT 2026

15

AT A GLANCE

Measuring whether AI outputs and workflows are dependable enough for their purpose.

A demo shows that an AI system can work. Evaluation shows how often it does, on what, and at what cost when it fails. This playbook explains how to define quality for an AI task, build test sets from real work, score outputs with people and automated judges, set pass marks, run human review and monitor after launch. It gives a release decision method that a business owner can use, with templates and illustrative numbers clearly labelled as examples.

WHO IT IS FOR Product owners, operations and quality leaders, analysts, engineers and business sponsors who must decide whether an AI workflow is ready to rely on.

TIME TO USE 70 minutes to read; one working session to draft your first test set

What you will have at the end

- A definition of quality for your AI task in terms a business owner can sign
- A starter test set design drawn from real cases, including hard and rare ones
- A scoring method that combines human judgment with automated checks
- A pass mark, release decision rule and post-launch monitoring plan
- A regression habit so improvements do not silently break what worked

Contents

01	Why impressive demos are not evidence	03
02	Matching the rigor of testing to the stakes	04
03	The anatomy of an evaluation	05
04	Building a test set that reflects real work	07
05	Metrics, pass marks and the release decision	08
06	Keeping quality after launch	10
07	Roles and ownership for quality	11
08	Setting up an evaluation capability in 90 days	13
09	Common failures and an illustrative scenario	14
10	A first quarter that produces evidence	15
11	Terms used in this publication	16

EXECUTIVE BRIEF

Why impressive demos are not evidence

01



AI output is variable by nature, so dependability has to be measured on many cases, not judged from a few good ones.

Traditional software is deterministic: the same input gives the same output, and a test either passes or fails. Language model systems are probabilistic. The same question may produce slightly different answers, and a change to a prompt, a model version or a source document can shift quality in ways nobody notices without measurement. A demo of five good examples tells you almost nothing about the sixth hundred.

Evaluation replaces opinion with a repeatable procedure. You collect representative cases, define what a good answer looks like, score the system against them, and decide whether the results are good enough for the stakes involved. You then repeat the procedure whenever something changes. The discipline is similar to quality assurance in manufacturing: sampling, specification, tolerance and corrective action.

- Quality is task-specific. A good summary for an executive differs from a good summary for a compliance file.
- The cost of errors is uneven. A missed detail in a meeting note is minor; a wrong dosage or contract term is not.
- A single accuracy number hides what matters. Break results down by case type and by error severity.
- Evaluation is continuous. Treat it as a standing capability, not a project gate.



PLAIN DEFINITION

Evaluation is a set of real test cases, a scoring rule, a pass mark and a record of results over time. If you cannot point to those four things, you are relying on impressions.



WORKING QUESTION

For the AI task you care most about, what evidence besides a demo would convince a skeptical colleague that it is dependable?

WHERE IT FITS

Matching the rigor of testing to the stakes

02

Test depth should follow the cost of a wrong answer and the degree of human review that follows.

Different tasks have different failure costs and different ways to check them. Some outputs have a single right answer, such as extracting an invoice number. Others are open-ended, such as drafting a customer reply, where judgment and style matter. Some feed directly into decisions, while others are always edited by a person. The right test design, pass mark and review level differ accordingly.

TASK TYPE	HOW TO CHECK	RIGOR NEEDED
Extraction (fields from documents)	Compare to a labelled answer key field by field	Medium to high; automated scoring works well
Classification and routing	Compare to expert labels; examine the confusion matrix	Medium to high; focus on costly misroutes
Question answering over documents	Check correctness, grounding and citation against sources	High; hallucination is the key risk
Drafting and summarizing	Human rubric scoring, with automated checks for facts and format	Medium; a human edits before use
Agent workflows with tool calls	Check outcome and each step, plus safety boundaries	High; errors become actions
Customer-facing responses	Rubric, red-team cases, live sampling	High; reputational and legal exposure

When evaluation is not the main answer

- If a task is rare and each case is unique, a test set may be too thin and a review process is the better control.
- If the stakes are very high and the AI only assists, the key measure may be whether the reviewer catches errors, not the AI's raw accuracy.
- If a simple rule or search solves the task, you may not need a model, and so you do not need this evaluation.

WORKING QUESTION

Which of your AI tasks has a checkable right answer, and which depends on judgment that needs a rubric?

HOW IT WORKS

The anatomy of an evaluation

03

Every evaluation has the same parts: a task definition, test cases, scoring and a decision rule.

01 Define the task and quality criteria

Write down the input, the expected output and the dimensions of quality: correctness, completeness, grounding in sources, tone, format, safety. Agree them with the business owner.

02 Build the test set

Gather real cases from past work, covering typical, hard and rare situations. Record the expected result or a rubric for each.

03 Run the system

Run every case through the current system version under realistic conditions, and store the outputs with the version identifiers.

04 Score

Score with trained reviewers, automated checks or both. Capture the reason for each failure, not only pass or fail.



05 Decide and record

Compare results to the pass mark, decide to release, fix or stop, and keep the record for later comparison.

Scoring methods

METHOD	STRENGTH	LIMIT
Exact or rule-based match	Cheap, objective, repeatable	Only fits tasks with a fixed answer
Human rubric scoring	Captures judgment and nuance	Slow, costly, reviewers disagree without calibration
Model-as-judge	Scales to many cases for open-ended output	Has its own biases; must be checked against human scores
Pairwise comparison	Good for choosing between two versions	Does not give an absolute quality level
Production sampling	Reflects real use	Arrives after launch; needs sampling discipline

CHECK YOUR JUDGE

If you use an AI model to grade outputs, have people grade a sample of the same cases and compare. Report how often the judge and humans agree. A judge you have not checked is another unmeasured system.

WORKING QUESTION

Who in your organization would agree on the rubric for a good answer, and have they ever scored the same ten outputs independently?

TEST SETS

Building a test set that reflects real work

04



A test set is your most valuable evaluation asset, so build it from real cases and protect it.

Start with the work, not the model. Pull actual past inputs from the process the AI will support: emails, tickets, contracts, forms, questions. Remove or mask personal data where required. Ask experienced staff to produce the correct output or to score outputs against a rubric. The cases should look like what arrives on a normal Tuesday, plus the ones that cause trouble.

What a good set contains

- Typical cases that make up most of the volume, so the headline result is meaningful.
- Hard cases: ambiguous inputs, messy scans, unusual formats, conflicting sources.
- Rare but costly cases: those where an error would cause real harm or a complaint.
- Cases where the right behavior is to refuse, ask for clarification or escalate to a person.
- Adversarial cases drawn from known misuse or attempts to confuse the system.

Size depends on how precise you need to be. A set of fifty carefully chosen cases can reveal gross problems and guide early iteration. It cannot prove a low error rate. If you need confidence that a serious error occurs in fewer than a small share of cases, you need far more cases, or you need to rely on review controls and monitoring instead. Ask a statistician or analyst to help size the set for your tolerance. Treat numbers such as an illustrative 200 cases as a starting point for discussion, not a rule.

KEEP A HOLD-OUT SET

Do not tune the system on the whole test set. Keep a portion unseen, or you will overfit and the results will flatter you. Refresh the set as the business, products and documents change.

Test set quality check

- Cases come from real work, not invented by the build team

- Expected answers were produced or checked by experienced staff
- Hard, rare and refusal cases are included on purpose
- Personal and confidential data is handled under approved rules
- A hold-out portion is protected from tuning
- Each case is tagged by type so results can be broken down

WORKING QUESTION

Which twenty real cases from last month would you put in the test set first, and who would write the correct answers?

METRICS AND RELEASE DECISIONS

Metrics, pass marks and the release decision

05

Choose a few metrics that reflect business risk, set the pass mark before testing, and decide with severity in mind.

Set the baseline first. Measure how the current human process performs on the same cases, in accuracy, time and cost. AI does not need to be perfect; it needs to be good enough relative to the alternative and to the control around it. Then agree the pass mark with the business owner before you run the test, to avoid moving the goalposts after seeing results.

METRIC	MEANING	USE WHEN
Accuracy	Share of outputs judged correct	Fixed-answer tasks with balanced outcomes
Precision and recall	Share of flagged items that are right and share of true items found	Detection and classification where misses and false alarms cost differently
Groundedness	Share of claims supported by the cited source	Question answering over documents

METRIC	MEANING	USE WHEN
Severity-weighted error rate	Errors counted by harm, not equal	Any task with unequal failure costs
Escalation rate	Share of cases handed to a person	Workflows with a human fallback
Time and cost per task	Effort and spend versus baseline	Business case and capacity planning



Split errors into severity classes. For example: critical, such as a wrong figure sent to a customer or a safety-relevant omission; major, such as a misleading but recoverable answer; minor, such as style. Set a different threshold per class. A common rule is zero tolerance for critical errors in the test set, a low ceiling for major errors, and a looser ceiling for minor ones. The actual numbers must come from your risk appetite, not from this page.

ILLUSTRATIVE RELEASE RULE (EXAMPLE ONLY)

Release if: no critical errors on the hold-out set; major errors below the ceiling agreed with the owner; performance not worse than the human baseline; and reviewer override tracking in place. Otherwise fix, narrow the scope or stop.



WORKING QUESTION

What is the one error type that would make you pull this system, and is it counted separately in your results?



HUMAN REVIEW AND MONITORING

Keeping quality after launch

06

Quality drifts, so build review, sampling and monitoring into operations from the first day.

Launch is the beginning of measurement. Inputs change, documents are updated, vendors change model versions and users find new ways to ask. Plan a staged release: limited users, then wider use, with tighter review at the start. Keep humans reviewing outputs in proportion to risk, and sample live traffic so you see real performance, not test-bench performance.

- Sample a fixed share of live outputs each week for scoring against the same rubric as the test set.
- Capture user feedback in the workflow, such as a one-click flag, and route flagged items to the owner.
- Track reviewer overrides and edits. Rising edits suggest declining quality; zero edits may suggest rubber-stamping.
- Log the model version, prompt version and source versions with every output so you can trace changes.
- Rerun the full test set whenever you change a prompt, model, retrieval setting or source collection.

01 Pilot with close review

Start with a small group and review most outputs. Compare live results to the test set.

02 Widen with sampling

As results hold, move to sampled review and set triggers for returning to full review.

03 Run regression

Before each change goes live, rerun the test set and compare to the previous version, case by case.

04 Set stop rules

Define in advance the signals that pause the system, such as a critical error or a drop below the pass mark.

VERSION CHANGES

A vendor model update can change behavior without any change on your side. Ask for notice, pin versions where possible, and run your regression set before adopting new ones.



WORKING QUESTION

If answer quality dropped ten percent next month, what in your current setup would notice?

OPERATING MODEL

Roles and ownership for quality

07



Evaluation works when the people who know the work own the standard, and the people who build the system do not grade their own homework.



Separate three responsibilities. The business owner decides what good means and what error rate is tolerable, because they carry the consequence. The builders improve the system and run the tests. An independent reviewer, from quality, operations or an analytics function, confirms the test set is fair and the results are reported honestly. In small teams one person may hold two roles, but the person who defines the pass mark should never be the person under pressure to ship.

ROLE	RESPONSIBILITY	TYPICAL HOLDER
Business owner	Defines quality, severity classes, Process or department head	pass mark, accepts residual risk
Subject-matter reviewers	Write expected answers and score outputs with the rubric	Experienced practitioners



ROLE	RESPONSIBILITY	TYPICAL HOLDER
Builder or product team	Runs tests, diagnoses failures, improves the system	Engineering or AI team
Evaluation lead	Owens test set integrity, scoring consistency and reporting	Quality, analytics or data lead
Operations owner	Runs live sampling, feedback and stop rules	Team lead

- Give reviewers protected time. Scoring done in spare minutes becomes careless.
- Rotate reviewers on a sample so you can measure agreement between them.
- Report results to the business owner in plain language, with the worst errors shown as examples, not just averages.
- Keep a change log that ties each quality movement to a cause: prompt, model, source or data.

GOVERNANCE LINK

Evaluation evidence is what a governance review needs for medium and high risk uses. Store test sets, scoring rules, results and release decisions where an auditor or reviewer can find them.

WORKING QUESTION

Who will tell you the truth about a failing result, and do they have the standing to stop a launch?

IMPLEMENTATION GUIDE

Setting up an evaluation capability in 90 days

08

II

Begin with one workflow and make evaluation routine before spreading it.

01 Choose the first workflow

Pick a task with real volume, clear stakes and an engaged business owner. Write the quality criteria on one page. 

02 Measure the human baseline

Score current human performance on a sample of cases for accuracy, time and cost. This is the bar to beat.

03 Assemble the first test set

Collect and label an initial set of real cases with typical, hard and refusal examples. Hold out a portion.

04 Calibrate reviewers

Have two or three reviewers score the same outputs, discuss disagreements and refine the rubric until agreement is acceptable. 

05 Run, fix, rerun

Evaluate, categorize failures, improve the system and rerun. Track results by version.

06 Decide and monitor

Apply the release rule, launch in stages and set up sampled review and a dashboard.

Release readiness

- Quality criteria and pass marks agreed in writing before testing
 - Results broken down by case type and error severity
 - No unresolved critical errors on the hold-out set
 - Human baseline measured for comparison
 - Review and escalation path defined and staffed
 - Monitoring, feedback capture and stop rules in place
- 

- Version logging enabled for model, prompt and sources

Tools



You can start with a spreadsheet, a shared rubric and a script that runs cases and stores outputs. Dedicated evaluation tooling helps once you have many cases and versions, but it does not replace the work of choosing good cases and agreeing quality. If you buy tools, ask whether you can export your test sets and results, so the most valuable asset stays yours.

WORKING QUESTION

Who is the single business owner who will sign the pass mark for your first workflow?



PITFALLS

Common failures and an illustrative scenario

09



Most evaluation failures come from testing the wrong thing, or testing once.

- Cherry-picked examples: the test set is built by the people who built the system.
- Tuning on the test set, so scores rise but real performance does not.
- One blended accuracy number that hides costly errors inside a good average.
- Judging with an unvalidated AI grader.
- Skipping rerun after a small prompt or source change.
- Treating human review as a guarantee without measuring reviewer quality.



ILLUSTRATIVE SCENARIO (HYPOTHETICAL)

A distribution company deploys an assistant to answer supplier terms questions. It scores well on the build team's 30 hand-picked questions. In live use, staff report wrong answers on questions about expired contract versions. A test set built from real tickets shows most failures involve superseded documents. The fix is source version control, a refusal rule and regression runs.



The scenario illustrates the typical gap between a demo set and real work. The cure is not a smarter model but a better view of failure. Categorizing failures by cause, such as outdated source, missing context, ambiguous question or model error, tells you whether to fix data, retrieval, prompts, scope or human review. Treat each failure as a diagnostic, and record it.

WORKING QUESTION

Which of your current AI claims is supported by a test set you did not build yourself?



30 / 60 / 90

A first quarter that produces evidence

10



In ninety days you can move one workflow from impression to evidence, and create the habit that makes the next one faster.

First 30 days: ground it

- Select the first workflow and write its quality criteria
- Measure the human baseline on a sample of cases
- Collect an initial set of real cases and agree expected answers
- Draft severity classes and a first pass mark with the owner



Days 31 to 60: test it

- Calibrate reviewers on a shared rubric
- Run the first evaluation and categorize every failure
- Fix the top causes and rerun by version
- Hold out a portion of cases and protect it

Days 61 to 90: decide

- Apply the release rule and launch in stages
- Start weekly sampled review of live outputs
- Add feedback capture and stop rules
- Make test set refresh and regression runs a standing routine

GLOSSARY

Terms used in this publication

11

TERM	MEANING
Test set	A collection of representative cases with expected answers or rubrics used to measure system quality.
Hold-out set	Cases kept unseen during tuning so they give an honest measure of performance.
Baseline	The measured performance of the current process, used as the bar a new system must meet.
Rubric	A written scoring guide listing criteria and what each score level means.
Model-as-judge	Using an AI model to score another system's outputs, validated against human scores.
Precision	Of the items the system flagged, the share that were truly positive.
Recall	Of all truly positive items, the share the system found.
Groundedness	The degree to which claims in an output are supported by the cited sources.

TERM	MEANING
Hallucination	A fluent output that is unsupported by sources or simply false.
Regression test	A rerun of existing tests after a change to confirm nothing that worked has broken.
Inter-rater agreement	How consistently different reviewers score the same output.
Severity-weighted error	An error measure that counts mistakes by their harm instead of equally.

A note on use

Numbers in this playbook are illustrative. Set pass marks from your own risk appetite and baseline measurements

TAKE THE NEXT STEP

Take the free Halden assessment at halden.company/assessments, or talk to us at halden.company/contact about turning this into a plan for your organization.