

Evaluating the Quality of Generative AI



Evaluation is now a core discipline, not an afterthought



Poor evaluation costs: hallucination, brand damage, learner misinformation



Silent regressions and compliance exposure follow weak quality checks



Generative quality is multidimensional: many valid answers, high variance



Roadmap: dimensions, human and automated methods, CI gates, governance



Why Generative AI Quality Is Different



Deterministic QA checks pass/fail; generative QA must accept varied, valid answers



Variance sources: temperature, model drift, retrieval, multi-step agent workflows



Quality is multi-dimensional and context-dependent; one score hides more than it reveals

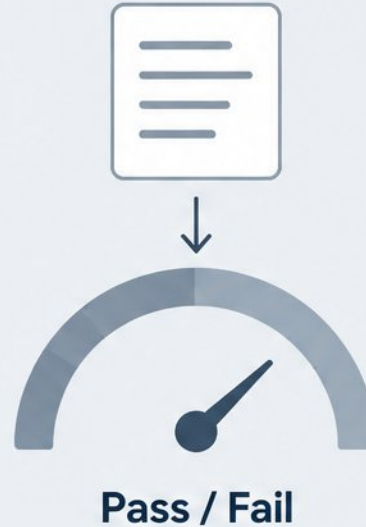


Test design shifts: sampling, rubrics, repeat runs, confidence intervals

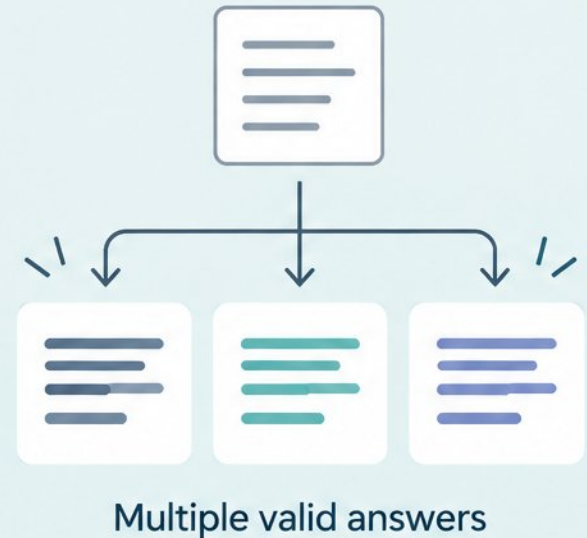


Keep development and final test data strictly separate

Deterministic QA



Generative QA



Quality Is Multi-Dimensional



One score hides more than it reveals

Core Quality Dimensions

Different dimensions, working together for helpful, trustworthy answers.



Correctness and factuality

verified claims, with hard gates on quantities, dates, entities

✓ Provides accurate claims with correct quantities, dates, and entities.

✗ Includes incorrect or unsupported facts, such as wrong dates or entities.



Grounding and faithfulness

every claim supported by retrieved or provided sources

✓ Each claim is supported by retrieved or provided sources.

✗ Makes claims without supporting sources.



Relevance, completeness, instruction following

answers the real need without gaps or padding

✓ Directly addresses the request, is complete, and follows instructions.

✗ Misses key parts of the request or adds unnecessary information.



Fluency, coherence, tone, audience fit

readable, logical, right for learners and customers

✓ Clear, coherent, and appropriately toned for the audience.

✗ Hard to follow, inconsistent, or wrong tone for the audience.



Safety, privacy, bias, compliance

no harmful or policy-violating content, including excessive refusal

✓ Provides a helpful response without harmful or policy-violating content.

✗ Includes harmful content, violates policy, or refuses unnecessarily.



Weights and mandatory criteria should reflect the service's purpose,
not a single overall score.

Human Evaluation and Rubric Design



Pairwise beats Likert: most reliable, minimizes scale bias; binary loses nuance



Analytic rubrics: unidimensional criteria, behavioral anchors, worked examples



Add a separate pass/fail threshold alongside the numeric score



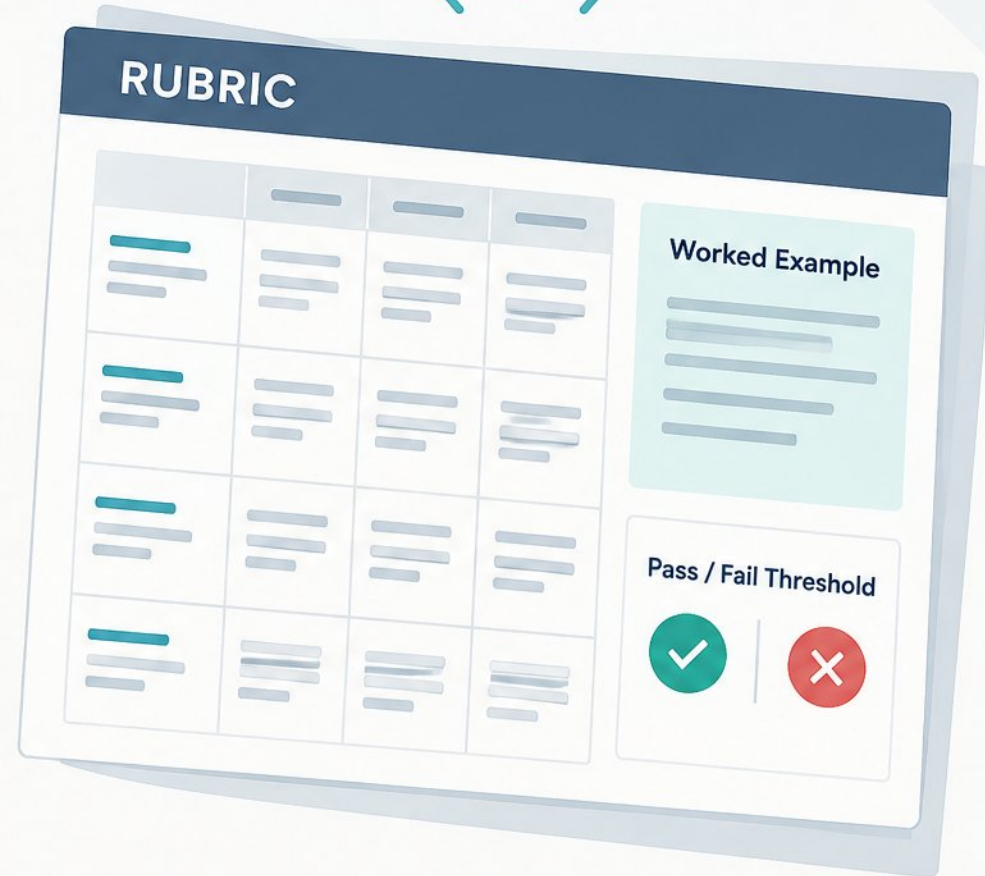
Qualify raters on 20-30 gold items; require 80% agreement



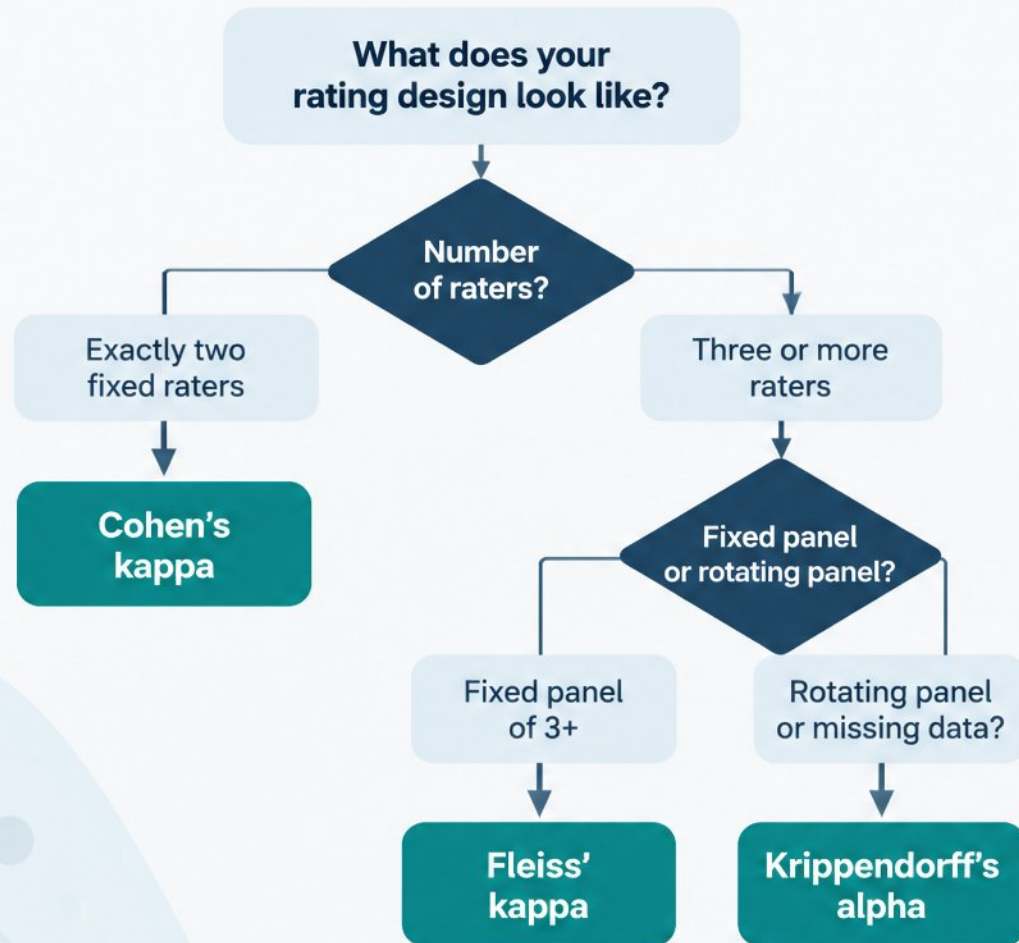
Pilot with 3-5 raters, 20-30 items; revise if kappa falls below 0.40



Control order, length, anchoring, and fatigue: randomize, cap sessions at 45 minutes



Measuring Whether Raters Agree



- Choose the metric from your rating design, not habit



- Cohen's kappa: exactly two fixed raters, categorical items



- Fleiss' kappa: fixed panels of 3+ on categorical items



- Krippendorff's alpha: ordinal scales, missing data, rotating panels



- Wrong coefficient = silent validity failure; ordinal needs ordinal distance



- Report one chance-corrected metric plus raw percent agreement



- Thresholds: 0.67 tentative, 0.80 strong; agreement is not validity



Automated Metrics and Their Limits

- BLEU, ROUGE, METEOR, BERTScore correlate weakly with human judgment
- They reward fluent but degenerate or repetitive output
- Negation, rare words, and reordering often go undetected
- Reference-free metrics and perplexity carry their own biases
- Use them as diagnostics and regression signals, not headline scores

HUMAN REFERENCE

The cat is not on the table.

She arrived before
the meeting started.

The model should be concise
and avoid repetition.

He saw a quokka
in the forest.

MODEL OUTPUT (PROBLEMATIC)

→ The cat is on the table.

→ Before the meeting started,
she arrived.

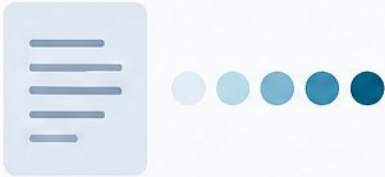
→ The model should be concise.
The model should be concise.

→ He saw a squirrel
in the forest.

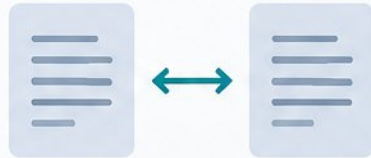


LLM-as-Judge: Design Patterns

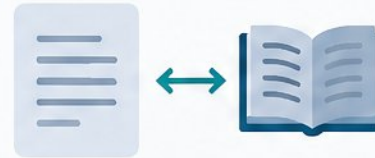
Pointwise scoring



Pairwise comparison



Reference-guided grading



Rubric-based binary verdicts



- Four core patterns: pointwise scoring, pairwise comparison, reference-guided grading, rubric-based binary verdicts
- Rubric-based evaluation is more granular and interpretable than holistic scores
- Rubrics cut position bias and score-calibration problems by removing direct output comparisons
- Design choices: chain-of-thought before scoring, explicit detailed rubrics, per-criterion calls
- Evaluate each criterion separately to reduce criterion conflation and halo effects
- Verdict-balanced few-shot examples improve judge reliability
- Scale design matters: 0-5 grading yields strongest human-LLM alignment



LLM-as-Judge: Biases and Calibration



Known judge biases: self-preference, position, verbosity, authority, score clustering.



Self-preference persists even with objective rubrics, shifting scores double digits.



Mitigate: cross-family ensembles, position swapping, length control, abstention when uncertain.



Calibrate against human anchors: Bayesian linear for small sets, non-parametric flow for large.



Linear probes give cheap uncertainty estimates; re-validate judges periodically.

From Biased Judgments to Calibrated Alignment



Identify Biases



Apply Mitigations



Calibrate to Human Anchors



Estimate Uncertainty



Re-validate Periodically

Hallucination, Grounding, and Safety Checks

A layered evaluation pipeline for more reliable and trustworthy AI outputs.



Detect hallucinations

- claim extraction
- source attribution
- self-consistency sampling
- external verification



Verify grounding

- citation accuracy
- quote fidelity
- retrieval-to-answer traceability
- abstain when evidence is thin



Screen safety

- toxicity and policy classifiers
- PII detection
- bias and fairness across demographic groups



Test adversarial robustness

- prompt injection
- jailbreaking
- guardrail evasion
- plus false positives causing refusals



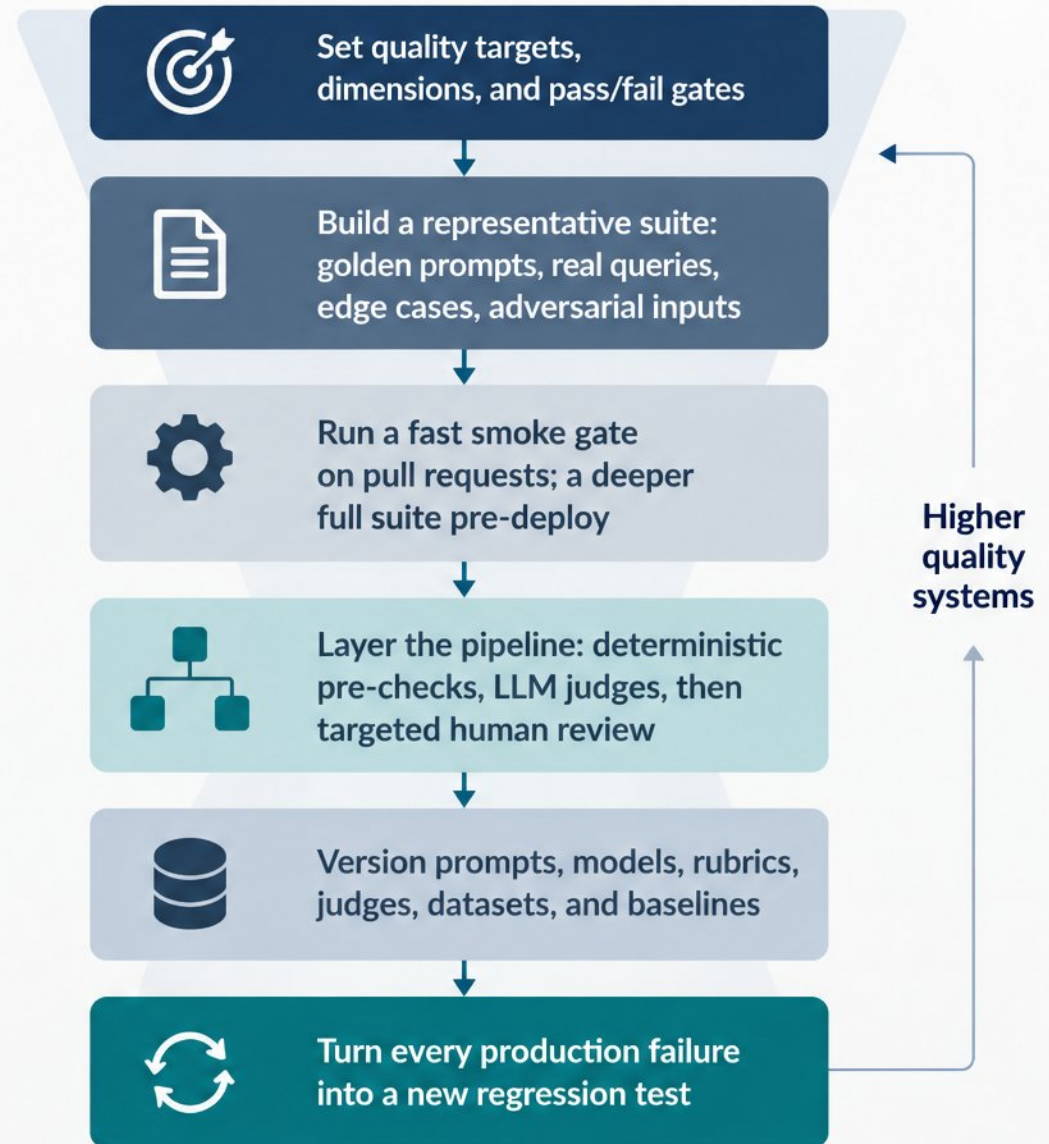
Design around error asymmetry

- a missed safety violation costs more than a flagged benign output

Building a Layered Evaluation Workflow

From quality goals to continuous improvement

- 1 Set quality targets, dimensions, and pass/fail gates before writing tests
- 2 Build a representative suite: golden prompts, real queries, edge cases, adversarial inputs
- 3 Run a fast smoke gate on pull requests; a deeper full suite pre-deploy
- 4 Layer the pipeline: deterministic pre-checks, LLM judges, then targeted human review
- 5 Version prompts, models, rubrics, judges, datasets, and baselines
- 6 Turn every production failure into a new regression test



Wiring Evaluation Into CI and Production



Treat quality like a build artifact:
declarative suites in version control



Gate on deltas, not absolutes:
“is this worse than it was?”



Post PR comments with per-case
delta tables showing regressions
and magnitude



Handle non-determinism: repeated
runs, statistical significance,
separate infra errors



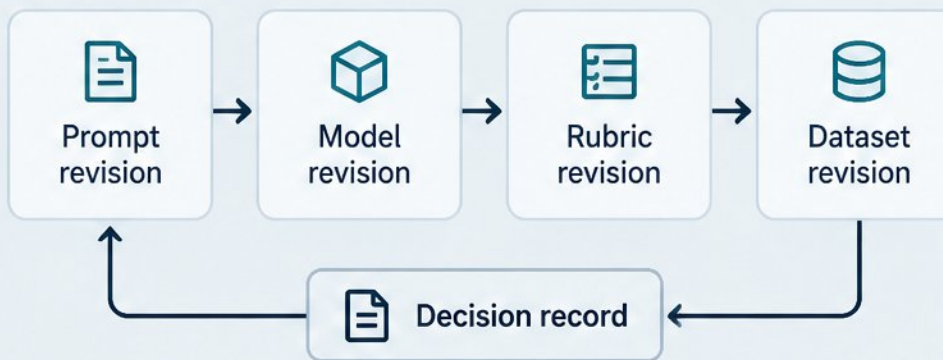
Monitor production for drift, then
feed captured traces back into
regression suites



Blocked Merge



Versioned evidence chain



Scorecards and Case Examples by Function



Content teams

- Content teams: accuracy, brand voice, SEO, publication readiness — human review as exception path



Educators

- Educators: curriculum alignment, age appropriateness, misconception risk; aggressive filtering rejects most raw items



Product teams

- Product teams: task success, user satisfaction, retention or deflection, latency, cost per interaction



QA specialists

- QA specialists: traceability, reproducibility, audit evidence, shared defect taxonomy



One shared scorecard reconciles conflicting stakeholder priorities

Governance, Transparency, and Audit Trails



Name who signs off on rubrics, evaluations, and releases



Write down the basis for each risk tier



Keep chronological, tamper-evident records linking provenance to decisions



EU AI Act Article 50 transparency duties apply from 2 August 2026



Label content consistently: human-authored, AI-assisted, AI-generated



Human review exempts public-interest text only if substantive



Prompt



Model



Rubric



Dataset



Decision record

Hands-On Practice and 30/60/90-Day Plan



Exercise: score outputs with a shared rubric, then compute inter-rater agreement



Exercise: draft an LLM-judge prompt, flag self-preference, position, verbosity, scale compression



Exercise: define one use case's dimensions, thresholds, mandatory gates, human review points

30



Golden set and smoke gate

60



Layered calibrated judges

90



Production drift monitoring

Checklist:



test set



rubric



automated checks



human review



versioning



CI gate



monitoring



audit trail



Takeaway: evaluation is continuous engineering, not one-time certification

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/f50a5bb1/public