

The LLM testing checklist

35 checks for an AI feature before it reaches real users. Tick what you already do. The gaps are where your risk sits.

1. Define what good looks like

- ☐ A scoring rubric for outputs, covering accuracy, relevance, tone and safety, with a clear description of each score.
- ☐ A short list of things the feature must never do, such as leak personal data, discuss banned topics or break the output format.
- ☐ A pass threshold the team has agreed, for example the share of golden-set outputs that must score well before release.

2. Build a golden set

- ☐ The awkward inputs: ambiguous questions, long or messy documents, edge cases from your domain.
- ☐ Every input that has gone wrong before, added the day it's found.
- ☐ The inputs that matter most to your customers, weighted by how often they happen.
- ☐ Small enough that someone actually reads the results. Versioned, so you know which set a result came from.

3. Keep hard checks where they belong

- ☐ Output structure: valid JSON or the required format, required fields present, length limits respected.
- ☐ No personal data, secrets or internal identifiers in outputs.
- ☐ Banned content and topics blocked, every time.
- ☐ Citations or source references present where the feature promises them.

4. Treat non-determinism as normal

- ☐ Run each golden-set input several times and look at the distribution of scores.
- ☐ Record the model settings that affect variation, such as temperature, for every test run.
- ☐ Fail the run on the share of poor outputs, not on any single difference in wording.

5. Test for misuse and prompt injection

- ☐ Instructions hidden in user input or uploaded documents ("ignore your previous instructions") are treated as content, not commands.
- ☐ Attempts to extract the system prompt, other users' data or confidential context fail.
- ☐ If the feature can call tools or take actions, every action is checked against what the user is allowed to do.

6. Check grounding and hallucination

- ☐ Answers stay within the source material and say so when the answer isn't there.
- ☐ Behaviour is tested with missing, outdated and contradictory documents.
- ☐ A sample of answers is checked against the source by a person, not only by another model.

7. Control model and prompt changes

- ☐ The model version is pinned wherever the platform allows.
- ☐ Any change to the model, prompt, settings or retrieval data re-runs the golden set against the last good baseline.
- ☐ Results and the reason for each change are recorded in release notes.

8. Keep people in the loop

- ☐ Someone reads a sample of real outputs on a regular schedule.
- ☐ If a model is used to judge outputs, its scores are checked against a person's scores before you rely on it.
- ☐ Users can flag a bad answer easily, and those flags reach the team.

9. Monitor in production

- ☐ Inputs and outputs are logged in a privacy-aware way, with personal data minimised.
- ☐ Alerts exist for spikes in flagged answers, refusals, errors, latency and cost.
- ☐ Problems found in production become new golden-set inputs.

10. Cover privacy and compliance

- ☐ You know what data is sent to the model provider, how long they keep it and whether they train on it.
- ☐ Users are told when they are dealing with AI or reading AI-generated text.
- ☐ You've checked whether the use case falls into a regulated or high-risk category, such as decisions about individual people.

11. Make the release decision

- ☐ The golden set meets the agreed threshold, and no hard check fails.
- ☐ Known weaknesses are written down, with a plan for each.
- ☐ Someone with authority has signed off on the remaining risk.