

PRACTITIONER · 6 MINUTE READ

Harnesses: everything around the AI that makes it reliable

The harness is everything around the AI model, such as instructions, skills, tools, checks and history, and it shapes how reliable your results are.

What it is

The AI model is the part that reads and writes. The harness is everything around it: standing instructions, tools, checks and the record of what changed.

Think of a car. The model is the engine. The harness is the rest of the car: steering, brakes, dashboard and seatbelts. A powerful engine without brakes is not a better car. Improving the harness can change everyday results as much as switching model.

A harness usually includes some of these parts:

- **Standing instructions:** a file the assistant reads every time, often written in Markdown (see guide 06).
- **Skills and commands:** saved ways of doing repeat tasks (see guide 11).
- **Connectors and tools:** safe access to your documents and systems (see guide 13).
- **Memory:** what the assistant keeps between conversations (see guide 24).
- **Hooks:** rules that run at a set moment, such as before a file is saved (see guide 25).
- **Guardrails and permissions:** what the AI may and may not do (see guide 26).
- **Evaluations:** tests you run before trusting a change (see guide 27).
- **Version history:** a record of every change to instructions and files, so you can see what changed and go back.
- **A learning loop:** a habit of recording mistakes and turning them into rules (see guide 24).

Why it matters

- The same model gives steadier results when it always gets the same instructions, tools and checks.
- Checks and permissions can catch many problems before they reach customers.
- Version history means a bad change can be undone quickly.
- When a new model arrives, your instructions, skills and tests carry over.

How to do it

1. Choose one repeat task where AI already helps.
2. Write one standing instructions file: purpose, audience, house style and what must never happen.
3. Save one skill for the step you repeat in that task.
4. Add one check: a short list a person ticks before the output is used.
5. Keep every version of these files, with a date and one line saying what changed.
6. Name an owner who approves changes and reviews the set-up on a fixed schedule, for example every three months.
7. Add more parts only when a real problem calls for them.

Example

Before: a team drafts customer replies with an approved assistant, each person with their own prompt. Tone varies, and one reply promises a refund nobody approved.

After: the team builds a small harness.

Standing instructions: customer_replies.md (v3)

- Audience: existing customers. Tone: warm, short, plain British English.
- Never promise refunds, discounts or dates. Write [needs approval] instead.
- Never include another customer's details.

Skill: "Reply draft" (v2), steps: read question, find answer in FAQ, draft reply.

Check before sending (person ticks):

[] Facts match the FAQ [] No promises [] Name and order number correct

Change log:

[date] v3: added "never promise dates" after a reply gave a delivery date.

A person still reads and sends every reply. When something goes wrong, the team updates one file and everyone gets the fix.

Common mistakes

- **Building everything at once.** Start with one file, one skill and one check.
- **No owner.** Files drift and contradict each other. Name an owner.
- **Changing instructions without a record.** Keep versions and a change note.
- **Thinking the harness replaces the person.** It reduces errors; a person still checks anything that leaves the team.

Try it this week

1. Pick one task you do with AI at least weekly (2 minutes).
2. Write a standing instructions file of no more than ten lines, using made-up examples only (6 minutes).
3. Write a three-item checklist for the output (3 minutes).
4. Save both with "v1" and today's date in a shared folder your team already uses (2 minutes).
5. Use them once and add one line to a change note (2 minutes).

Going further

- **Version control:** a system that keeps every saved version of a file, who changed it and why. Software teams use dedicated version control tools; many shared document platforms also keep version history. Either way, you can compare two versions and restore an older one.
- Keep harness files as plain text or Markdown, so people and assistants can read them and history shows which line changed.
- Assistants such as Claude, ChatGPT, Microsoft Copilot and Gemini offer projects, saved instructions or custom assistants; these are natural homes for the first parts of a harness.
- Connectors may use MCP (Model Context Protocol, an open standard for connecting AI to tools and data); review what each one can reach (see guide 13).
- Rerun your evaluations after any change (see guide 27).

Key words

- **Model:** the AI engine that reads and writes text.
- **Harness:** everything around the model that guides, limits and checks it.
- **Standing instructions:** a file the assistant reads at the start of every task.
- **Version history:** a saved record of every change, so you can compare and go back.
- **Owner:** the person who approves changes to a part of the harness.