

---

# Testing Agent

Technical notice

Version 0.8.3

28 October 2026

Moamen Elmasry

Free · [moamen.fr/en/testing-agent/](https://moamen.fr/en/testing-agent/)

# Contents

---

## Summary

### 1. The problem

### 2. The method

### 3. Architecture

### 4. Data and files

### 5. Rules and human decisions

### 6. Azure DevOps

### 7. Installation and uninstallation

### 8. Limits and recommended use

### 9. Version history

## Appendix. The eight skills

# Summary

---

The Testing Agent is an MCP server: a small program that the AI assistant in your editor calls to use tools. It equips the assistant to test: a tester's method, tools and rules, applied to a team's user stories, including those where part of the output is generated by AI. The assistant becomes a tester, and a guide for testers.

For each criterion of a user story, it tells what is tested against an exact result (classic code), what is measured on a reference set (generated answers) and what mixes both. It proposes the cases by phase (component, integration, system, AI evaluation), writes and runs code tests, reads JUnit reports and Azure DevOps builds, walks screen tests step by step, prepares the evidence and the bug draft, and keeps the tracking as text files in the project's repository. A local web interface shows each person what waits on them, walks them through their tests step by step and keeps their screenshots as evidence.

Approving a case, setting a result, accepting a deviation and deciding a Go / No Go remain human acts. The tracking register rejects certain writes and records every decision in the name of the person who takes it.

| At a glance |                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Type        | Local MCP server, in Python 3.10 or later                                                                                                    |
| Works with  | Installation planned for GitHub Copilot, Claude Code, Codex and Cursor                                                                       |
| Tracker     | Azure DevOps and Azure Test Plans, or none                                                                                                   |
| Data        | It stays on the machine. At run time, the Testing Agent opens no connection to the outside.                                                  |
| Price       | Free during the test phase; if it becomes paid, 30 days of notice, and nobody pays without having agreed to it. No resale or redistribution. |
| Support     | A single author, no commitment                                                                                                               |
| Version     | 0.8.3, 28 October 2026                                                                                                                       |

## Good to know before adopting it

- Designed for trials and team use. Have your IT department approve it before a deployment.
- It does not check a person's identity: the record relies on what they declare.
- The files are not signed: when installing by hand, Windows may show a warning or block the file (chapter 7).
- The code repository is not public.

## 1. The problem

A user story often mixes several kinds of criteria. Some are tested: an exact result is expected (a link expires after thirty minutes). Others are measured: several generated answers are acceptable, and only an evaluation on a reference set can judge them (a summary lists the decisions made). In a hybrid product, both coexist in the same user story.

Classic test tools expect an exact result and can only handle the first kind. For the second, the team then sets a threshold at random, or judges by eye, with no record, or blends both into an average that hides a failure.

In the author's practice, acceptance testing of an AI product rested almost entirely on one person: they sorted the cases by hand, created them in the tool, ran most of the tests and kept the method to themselves. The Testing Agent puts that method within reach of the whole team, in the assistant everyone already uses. A skill alone is a text of instructions; here, a local server acts on the project's files, keeps the state and refuses some writes. The assistant stays itself; its ability to test is augmented, the way an exoskeleton augments its wearer's.

*This observation comes from the author's experience. It has not been the subject of a survey.*

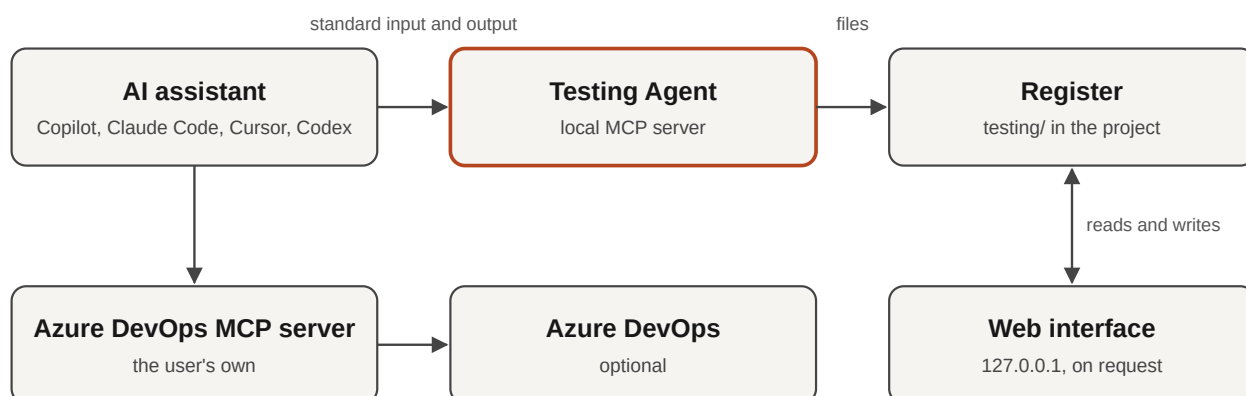
## 2. The method

The method is written in 21 notes, shipped in the archive (in English). It is based on the author's practice and on the classic test levels. It does not claim any standard.

| Notion                           | Content                                                                                                                                                                                      |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Definition of Done</b>        | What is tested, against an exact result. A flaky test counts as failed.                                                                                                                      |
| <b>Definition of Trustworthy</b> | What is measured: several answers are acceptable, judged on an annotated reference set, against a threshold approved by a person. Without an approved threshold, the verdict is "not ready". |
| <b>Deterministic</b>             | One correct result, checked exactly.                                                                                                                                                         |
| <b>Probabilistic</b>             | Several acceptable answers, measured on a reference set. Vetoes stop a critical error, slices stop an average from hiding a group: a good average never cancels a critical failure.          |
| <b>Hybrid</b>                    | The exact part is tested exactly, the generated part is measured on its own.                                                                                                                 |
| <b>Four phases</b>               | Component, integration, system, AI evaluation. The register also tracks an acceptance level.                                                                                                 |

A hybrid product is accepted only when every applicable gate passes independently: the Definition of Done, the Definition of Trustworthy and, when it applies, acceptance by users. The results are never blended into a single score. An AI judge, when used, must itself be tested, calibrated and versioned.

### 3. Architecture



The Testing Agent talks to the assistant through standard input and output (the MCP protocol): it listens on no port. Only the web interface, started on request, listens on 127.0.0.1. It reads and writes files in the project.

It does not call Azure DevOps itself: the assistant does, with the user's Azure DevOps MCP server, and the Testing Agent prepares or records what relates to it.

#### The local web interface

Started on request by the assistant, it lets people see and decide for themselves. Every write there goes through the same rules as the agent's, in the name of the person who entered it.

- Home: what waits on you now, most urgent first, what waits on others, and the campaign's progress by phase.

- Run a test: one step at a time, with the exact values to type ready to copy, then As expected, Not as expected or Blocked. A screenshot is pasted or dropped as evidence.
- What the agent did: every write by the agent with its reason, and what a person should review.
- Test cases, with a search and the approval of several cases at once; anomalies, actions, coverage, progress, and a history that reads as sentences.
- Export to Excel: one sheet per subject (cases, results, steps, defects, actions, requirements), to sort, filter and share.
- Usable with the keyboard alone, in light or dark mode, in English or French. A test in progress resumes after a refresh.

The Home page features a navigation bar with links: Home, Test cases, Defects, Actions, What the agent did, Coverage, Progress, History, and Help. Below the navigation bar, a greeting 'Hello Moamen' is followed by the status 'Test iteration 1: 2 of 6 cases run, 4 to go.' A section titled 'Yours now' contains three task cards. The first card, 'Run CT-002 in test late since 15 October 2026', has buttons 'Open the case' and 'Mark as done'. The second card, 'Qualify the failure of CT-004: defect, gap, test or environment issue, or settlement by the decision owner', has an 'Open the case' button. The third card, 'Run CT-001: Validate a summary system · Very high · US 4512 · In progress', has a 'Run' button.

The home page: what waits on you now, then what waits on others.

The test step interface for 'Run CT-006: Ask the assistant two questions, then save a note' shows the system 'High · EX-003' and dataset 'Edge values'. A progress bar indicates 'All values of the dataset "Edge values"'. The step is 'Step 4 of 4'. The instruction is 'DO Type user.test in Identifier and Line 1 | more Line 2 in Note, then Save'. Below this, a list of values to type is shown: '1. identifier' with the value 'user.test' and a 'Copy' button; '2. note' with the value 'Line 1 | more Line 2' and a 'Copy' button. A note explains: 'Starts or ends with a space, which is part of the test: copy the value rather than retype it.' and 'On several lines: copy it as is.' The 'EXPECTED RESULT' section shows 'the message Note saved is shown'. The 'TO COMPARE' section shows 'expected\_message' with the value 'Note saved' and a 'Copy' button. At the bottom, there are three buttons: 'As expected' (checked), 'Not as expected', and 'Blocked'. A link 'Not applicable' is also present.

A test step by step: the values to type, ready to copy, then the verdict.

## The tools exposed to the assistant

### Onboarding, profile and method

| Tool                      | Purpose                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| onboarding_status         | Where onboarding stands: profile, project, Azure DevOps connection, path, guide.                                                                                                                                                                                  |
| onboarding_ado_connection | Connects onboarding to the Azure DevOps MCP server; changes the configuration only with the person's agreement.                                                                                                                                                   |
| onboarding_save_profile   | Saves the person's name, role and language in their local profile.                                                                                                                                                                                                |
| tracking_settings         | Saves the project's settings: language, path, decision owner, Azure DevOps.                                                                                                                                                                                       |
| get_guide                 | Gives the user guide, in French or English, limited to the person's role.                                                                                                                                                                                         |
| list_skills               | Lists the eight skills and what each is for.                                                                                                                                                                                                                      |
| get_skill                 | Gives a skill's procedure and the list of its references.                                                                                                                                                                                                         |
| read_skill_file           | Reads a reference, template, schema or example of a skill.                                                                                                                                                                                                        |
| read_methodology          | Reads a note of the method.                                                                                                                                                                                                                                       |
| update_check              | Says whether a newer version exists, what it changes, keeps and replaces, checked every 7 days without disturbing: the assistant reads a public file on moamen.fr, nothing is sent, the Testing Agent never connects; installed only with the person's agreement. |

### Register and tracking

| Tool                    | Purpose                                                                                            |
|-------------------------|----------------------------------------------------------------------------------------------------|
| tracking_init           | Creates the tracking register (testing/ folder) in a project.                                      |
| tracking_upsert         | Creates or updates a tracked item: scope, requirement, test case, campaign, defect, action.        |
| tracking_status         | State of a campaign: coverage, progress, what is left, what waits on a person.                     |
| tracking_next_step      | Gives the one next action for a user story or the campaign, or the question to ask a named person. |
| tracking_render         | Regenerates the status file (Markdown) and the offline dashboard (HTML).                           |
| tracking_open_dashboard | Starts the local dashboard (127.0.0.1 only) and gives its link.                                    |

## User story, context and acceptance criteria

| Tool                   | Purpose                                                                                                                                   |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| tracking_story_sync    | Records a user story read from Azure DevOps with its parents, links and comments, splits its acceptance criteria and spots what changed.  |
| tracking_story_context | Records the agent's reading of the context (constraints, out of scope, inconsistencies, implicit requirements), for a person to validate. |
| tracking_story         | Shows a user story's sheet or its closure dossier; records a person's validation.                                                         |

## Code tests and screen tests

| Tool                   | Purpose                                                                                  |
|------------------------|------------------------------------------------------------------------------------------|
| summarize_test_results | Summarises JUnit XML reports by level: component, integration, system.                   |
| tracking_import        | Imports automated results: a JUnit XML report or an evaluation report.                   |
| tracking_case_brief    | Prepares a case run: steps with the dataset's values, expected results, history.         |
| tracking_record_result | Records one run of an approved case, step by step, in the name of the person who ran it. |

## AI evaluation

| Tool             | Purpose                                                                              |
|------------------|--------------------------------------------------------------------------------------|
| validate_dataset | Validates a reference set (JSONL): schema, duplicates, slice coverage.               |
| check_readiness  | Checks that an evaluation manifest is ready, without running it.                     |
| run_evaluation   | Runs an approved evaluation manifest: cases, evaluators, vetoes, slices, aggregates. |

## Azure DevOps

| Tool                      | Purpose                                                                                                                          |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| ado_diagnose              | Explains an Azure DevOps refusal (rights, Test Plans access level, tool not visible) and what the person can do.                 |
| tracking_ado_checklist    | Generates a user story's test checklist for Azure DevOps, acceptance criterion by criterion; only an OK is ticked.               |
| tracking_ado_testplan     | Prepares the publication of a campaign to Azure Test Plans: plan, suites, cases.                                                 |
| tracking_ado_suite        | Creates a user story's requirement-based suite through the REST API, with the person's Azure CLI sign-in, after their agreement. |
| tracking_import_ado_plan  | Loads a plan published in Azure Test Plans into the register.                                                                    |
| tracking_import_ado_build | Records the automated results of a build, read from Azure DevOps.                                                                |

| Tool                            | Purpose                                                                                                                    |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| tracking_import_ado_results     | Compares the results read from Test Plans with the register: records, confirms or reports a conflict, overwriting nothing. |
| tracking_publish_results        | Lists the results to set in Test Plans, each with its person; published only at their request.                             |
| tracking_mark_published         | Records that a result was set in Test Plans at the request of the person who ran the test.                                 |
| tracking_validation_note        | Prepares the evidence note to paste into Test Plans and the outcome to set there.                                          |
| tracking_declare_ado_validation | Records, in the person's name, that they set a result in Test Plans.                                                       |
| tracking_bug_draft              | Drafts an Azure DevOps bug from a failure, to be read before it is created.                                                |

## 4. Data and files

| Location                                                                          | Content                                                                                                                                  |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| %LOCALAPPDATA%\TestingAgent\<version> (Windows) or ~/.testing-agent/app/<version> | A copy of the Testing Agent, with its own Python environment.                                                                            |
| testing/data/ in the project                                                      | The tracking's source of truth: one JSONL file per kind of item, a project file and the log of every write.                              |
| testing/ÉTAT DES TESTS.md and testing/tableau-de-bord.html                        | Generated views, regenerable at any time.                                                                                                |
| testing/preuves/ in the project                                                   | Screenshots and logs attached to results: images, PDF or text, checked by their content, 3 MB at most each. Versioned with the tracking. |
| .vscode/mcp.json, .mcp.json, .cursor/mcp.json or ~/.codex/config.toml             | The testing-agent entry in the chosen assistant's configuration. Other servers are not modified.                                         |
| ~/.testing-agent                                                                  | The profile: name, role, language.                                                                                                       |

### The log of every write

Every write is recorded in testing/data/journal.jsonl: who (name and kind, human or agent), when, which item, and each field with its value before and after. These are text files: they are versioned with the code, compared and merged like it. When two people record at the same time, each in their own copy, the Testing Agent settles the merge on opening: nothing is lost, and an item that is renumbered is renumbered in the log.

### Network

- The Testing Agent opens no connection to the outside and sends nothing to its author.
- The web interface only listens on 127.0.0.1, rejects requests whose host or origin is not local, and cannot be shown inside another site.

- The installation downloads two Python dependencies (PyYAML and jsonschema) from the public PyPI index, into an environment of the Testing Agent's own.
- The Testing Agent asks for no key or token and stores none. Azure DevOps is called by the user's Azure DevOps MCP server, with its own permissions.

## Personal data

The tracking can contain names and observations entered by users. It stays in the project's repository, under that repository's access rules.

# 5. Rules and human decisions

---

## What the register refuses

Enforced by the code: a write that breaks the rule is rejected.

- Closing a defect without a passing retest.
- Approving a case whose expected result is still "to confirm".
- Assigning an action to the agent rather than to a named person.
- Taking a Go / No Go on a campaign that is not finished.
- Recording a write without an author.

## The instructions given to the assistant

Followed by the assistant, not enforced by the code.

- Do not invent a threshold or an importance.
- Do not set a result in Azure Test Plans in the person's place.
- Create a bug in Azure DevOps only with the person's agreement.
- Mark a manual test as passed only if a person says they ran it.

## Who decides

| Decision                                          | Who                                                                                              |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <b>Approve a test case</b>                        | The test decision owner. A case approved by the person who runs it is flagged for a peer review. |
| <b>Declare that a manual test was run</b>         | The person who ran it                                                                            |
| <b>Set a result in Azure Test Plans</b>           | The person who ran the test, with their own account                                              |
| <b>Accept a failure</b>                           | The decision owner, stating the accepted deviation and the reason                                |
| <b>Validate the business severity of a defect</b> | A person on the business side                                                                    |
| <b>Decide a Go / No Go</b>                        | The named decision owner                                                                         |

## Limit: identity

The server does not check the person's identity. A decision is recorded in the name of the person the assistant declares, once they have taken it. For an identity check, rely on the repository (history, access rights) and on Azure DevOps, where the result is set by the person themselves, under their account. An assistant can also depart from an instruction: the log keeps what was written, by whom, with the state before and after.

## 6. Azure DevOps

---

Azure DevOps is the team's shared reference. The Testing Agent does not call it itself: the assistant does, with the official Azure DevOps MCP server, the user's own.

### What it reads

- Projects and iterations: the current sprint is proposed, the person confirms.
- User stories and their acceptance criteria.
- Existing test plans, to join a plan instead of creating one.
- The automated results of a build.

### What it prepares, and writes only after the person's agreement

- One test plan per sprint, one suite per user story, one test case per dataset, with the phase in the title and a link to the story.
- A "Test tracking" section in the user story, regenerated after every result.
- A bug from a failure: reproduction steps up to the failing step, expected, observed, proposed severity.

### What it does not do

- Delete anything in Azure DevOps.
- Set a result in Azure Test Plans: the person does, with the evidence note the agent prepares for them.

These agreements are part of the instructions given to the assistant (section 5). Creating plans in Azure Test Plans needs a Basic + Test Plans licence. With Basic alone, you run tests without creating them. Without Azure DevOps, user stories are given as text and everything else works locally.

## 7. Installation and uninstallation

---

### With your assistant (recommended)

Copy the sentence on the Testing Agent page, [moamen.fr/en/testing-agent/](https://moamen.fr/en/testing-agent/), then paste it into the assistant, in Agent mode, in the project to equip. The assistant downloads [moamen.fr/downloads/testing-agent.zip](https://moamen.fr/downloads/testing-agent.zip), checks its SHA-256 checksum, extracts it to a temporary folder and follows the archive's "For your assistant" file: it checks for Python 3.10 or later and offers to install it if missing, runs the installation, then says how to turn the Testing Agent on in the tool. It announces every command, which the person approves.

### If the assistant cannot

- Download the archive: [moamen.fr/downloads/testing-agent.zip](https://moamen.fr/downloads/testing-agent.zip).

- Extract it: right-click then "Extract All" on Windows, double-click on a Mac.
- Open a terminal in the "Testing Agent/Fichiers - Files" folder and run `py -3 "mcp-server\wizard.py"` (Windows) or `python3 "mcp-server/wizard.py"` (macOS, Linux). Python 3.10 or later is needed.
- Answer two questions: the assistant you use, then the project. The installation ends with the steps to get started.

The archive holds no launcher script (.cmd, .command): installation goes through the assistant or through Python. The installer copies the Testing Agent to a permanent folder, creates a Python environment, installs two dependencies, configures the chosen assistant for the project, checks that the server answers and shows the steps to get started. If the configuration file contains comments, it is not rewritten: the entry to add is shown.

### Check the file, install without a window

The SHA-256 checksum published on the page compares with `Get-FileHash testing-agent.zip -Algorithm SHA256` (Windows) or `shasum -a 256 testing-agent.zip` (macOS, Linux).

From the command line: `py -3.12 "<folder>\Fichiers - Files\mcp-server\install.py" --client vscode --project "<project>"` (clients: vscode, claude, cursor, codex, other).

### Update

Give the page's sentence to the assistant again, or run `wizard.py` from the new version. The tracking registers and the profile stay as they are.

### Uninstall

Remove the testing-agent entry from the assistant's configuration and delete the Testing Agent's permanent folder. The tracking registers stay in the projects; the profile is deleted with `~/testing-agent`.

## 8. Limits and recommended use

---

- Designed for trials and team use. Have your IT department approve it before a deployment.
- A person's identity is not checked (section 5).
- The instructions given to the assistant depend on it: behaviour varies with the model used.
- Other MCP clients can be configured by hand; this is not verified.
- Azure DevOps is the only tracker supported. Jira and Monday are not.
- The files are not signed.
- The code repository is not public. Free, no resale or redistribution.
- A single author, with no guaranteed support or update schedule.
- The method is the author's own; it does not claim any standard.

## 9. Version history

| Version | Date       | Change                                                                                                                                                                                                                                                                                                |
|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0.8.3   | 2026-10-28 | Company HTTPS inspection (Netskope, Zscaler): publishing to Test Plans works again with Python 3.13 and later, certificates checked as Windows checks them.                                                                                                                                           |
| 0.8.2   | 2026-10-28 | Test phase: free for the whole phase, access from your space on moamen.fr; if it becomes paid, 30 days of notice, and nobody pays without having agreed to it.                                                                                                                                        |
| 0.8.1   | 2026-10-23 | Updates: a silent check every 7 days; the agent says whether a newer version exists; the installer keeps the data, copies it first and keeps the previous version until confirmed.                                                                                                                    |
| 0.8.0   | 2026-10-23 | The user story read in full (description, acceptance criteria, parents, comments); tracking criterion by criterion, changes spotted; requirement-based suites in Test Plans; results published to and read back from Azure DevOps; guided next step; refusals explained; tables that sort and filter. |
| 0.7.7   | 2026-10-21 | Installation by the assistant only: the double-click scripts, wrongly flagged by an antivirus, are removed.                                                                                                                                                                                           |
| 0.7.6   | 2026-10-20 | Search in test cases, approval of several cases at once, export of the tracking to Excel.                                                                                                                                                                                                             |
| 0.7.5   | 2026-10-20 | Web interface redesigned: a home page per person, step-by-step runs with values to copy and attached screenshots, what the agent did and why; team work through git; accessibility.                                                                                                                   |
| 0.7.4   | 2026-10-20 | Installation by the assistant, from a sentence to copy; archive arranged in four items, with readable names.                                                                                                                                                                                          |
| 0.7.3   | 2026-10-20 | Rights notices in the method and the skills; terms of use extended to the method.                                                                                                                                                                                                                     |
| 0.7.2   | 2026-10-20 | Archive focused on applying the method: 21 notes.                                                                                                                                                                                                                                                     |
| 0.7.1   | 2026-10-17 | Installation with a double-click, on Windows and Mac, with a permanent copy of the Testing Agent.                                                                                                                                                                                                     |
| 0.7.0   | 2026-10-17 | English and French: onboarding, guide, dashboard, status file, checklist, evidence note, bug draft.                                                                                                                                                                                                   |
| 0.6.1   | 2026-10-17 | Without Azure DevOps, onboarding no longer asks about Azure DevOps or the Test Plans licence.                                                                                                                                                                                                         |
| 0.6.0   | 2026-10-17 | Free. Installation for any agent-mode MCP assistant. Path with no tracker.                                                                                                                                                                                                                            |
| 0.5.0   | 2026-10-16 | Aligned with the official Azure DevOps MCP server; reading the results of a build.                                                                                                                                                                                                                    |
| 0.4.0   | 2026-10-16 | Onboarding connected to Azure DevOps: projects and the sprint are read instead of typed.                                                                                                                                                                                                              |
| 0.3.0   | 2026-10-16 | Phases per user story, datasets, settlement, Azure DevOps checklist.                                                                                                                                                                                                                                  |

## Appendix. The eight skills

| Skill                             | Purpose                                                                               |
|-----------------------------------|---------------------------------------------------------------------------------------|
| <b>audit-ai-evaluation</b>        | Audits an evaluation: is it properly governed?                                        |
| <b>build-golden-dataset</b>       | Builds the annotated reference set.                                                   |
| <b>design-ai-evaluation</b>       | Designs the evaluation of an AI feature: criteria, reference set, thresholds.         |
| <b>design-software-tests</b>      | Designs component, integration and system tests.                                      |
| <b>prepare-ai-testing-session</b> | Turns a feature, a user story, requirements and risks into a testing-session package. |
| <b>run-product-evaluation</b>     | Runs an approved evaluation against the product.                                      |
| <b>select-ai-evaluators</b>       | Chooses the evaluators and the metrics.                                               |
| <b>track-test-campaign</b>        | Tracks a campaign: register, dashboard, decisions.                                    |