AI TESTING LAB STUDY, PART TWO

AI OPERATOR **BOOKLET**

15 rules for working effectively with an AI model in testing

BASIS FOR THE FINDINGS

Each rule on this list reflects a specific pattern observed in the repositories of Sii Testing Lab participants. Sixteen teams working with AI tools searched for nineteen intentionally seeded defects in a loan application processing system. We reviewed around 350 bug reports, checking each one against the code, participant documentation and the OpenAPI specification. A typical team detected around six of the nineteen defects, while none of the teams found the two defects that would have been most costly in production. Engineering skill levels were comparable and the tools were the same, so the differences came down to how the teams worked. The practices that repeatedly appeared in the more effective teams and were missing from the others formed the basis of this list.

HOW TO USE THIS BOOKLET

Use this list in three ways: as checkpoints in the definition of done for a testing task, as questions during code review and as onboarding material for anyone starting to work with the model. The rules are technology-stack agnostic. We grouped the first eleven under the five quality gates described in the report, while the final four cover practices that apply throughout the entire workflow.

GATE 1

Independent correctness oracle

Derive the expected result from the specification, never from application behavior.

1. Build an independent oracle based on the specification, not on the application's behavior alone.
2. If computing the full formula is too costly, use a metamorphic invariant.

GATE 2

Assertions at the right layer

Money, statuses, permissions and decisions live in the API.

3. For business-critical values, make assertions through the API, not just the UI.

GATE 3

Separate adversarial pass

Coverage confirms that things work. Finding defects requires a separate pass.

4. Once basic coverage is in place, run a separate adversarial pass.
5. Turn every risk into a concrete scenario that can demonstrate a defect.
6. Write tests that can prove the system wrong, not just confirm the happy path.

GATE 4

Response hygiene & security

A review separate from domain testing and the least expensive to implement.

7. Run a brief security and response-hygiene sweep: sensitive fields in responses, endpoint authorization, CORS consistency, garbage input validation and the scope of data visibility across roles.

BRAMKA 5

Verify before you trust

A green result is a claim to falsify.

8. Do not let AI weaken assertions just to make a test pass.
9. Every bug should have a red-by-design repro or an expected-failure test.
10. Treat a failing test as a hypothesis of a defect rather than automatically classifying it as flaky.
11. Re-verify results using a separate agent, a separate session or a human reviewer.

CROSS-CUTTING PRACTICES

Workflow discipline & measurement

They apply regardless of which gate you are currently working through.

12. Maintain persistent context for AI in a file such as CLAUDE.md, AGENTS.md or MEMORY.md.
13. Log decisions: what AI suggested, what you rejected and why.
14. Do not build a framework at the expense of bug hunting.
15. Do not use test count as your primary measure of success.

Before you consider a test done, invert the expected value and run it again. A test that still passes with the wrong value is not guarding anything. This is the fastest way to check whether an assertion has teeth.

WHAT ARE THESE FINDINGS BASED ON?

The full study, including the data, charts and mechanisms behind each gate, is presented in the report "AI Testing Lab Study, Part II: AI Is Not Enough." It includes, among other things, a risk curve showing that the more costly a defect would be in production, the less often it was detected, as well as a comparison of reporting costs showing that the same outcome could cost seven times more or seven times less depending on the way of working.

How many of your "green" tests **would catch a bug headed for production?**

We will analyze your application using the same methodology we applied in the AI Testing Lab study. The results will show which risk areas are effectively covered by tests and where gaps still require attention.

Contact with experts

Iuliia Toporova
Business Development Manager