

Changing the Model Means Retesting the Workflow

By Junior Williams

Draft for review · 14 October 2026

A model change deserves the same care as a change to any other component that helps run a business process. The new version may produce better answers in a demonstration. You still need to establish what happens inside your workflow.

Think about replacing a component in a machine. Even when the replacement fits, someone needs to check that the machine still does its job. With AI, that job may involve finding information, applying rules, using tools and knowing when to ask a person for help.

The useful question is whether the updated workflow still performs the work it is authorized to do, under the conditions it will actually encounter.

Start with what the workflow must preserve

Consider a hypothetical assistant that helps staff answer customer questions. It retrieves approved guidance, drafts a response and sends uncertain cases to a reviewer. Some versions might also update a case record.

For that workflow, a good answer is only part of the requirement. The assistant must use information the employee is allowed to access, preserve the distinction between guidance and customer facts, and leave consequential actions at the agreed approval point.

Write those requirements down as testable behaviours. “Respect permissions” is a useful principle. A concrete test is that a user assigned to one account cannot retrieve another account's restricted information through the assistant.

The access control itself should be enforced by the application and connected systems. Model instructions alone should not carry that responsibility.

Test the whole path

A small regression set can reveal changes that an attractive demonstration misses. A regression test repeats a known case to check whether previously working behaviour still holds.

Include ordinary requests and cases that exercise the boundaries. For the assistant above, I would begin with these:

- **Source use:** Does the answer use the current approved guidance and identify where it came from? When two sources conflict, does the workflow handle that conflict as intended?
- **Access:** Does each test user see only the information their role permits? Repeat the check through retrieval and tool results.
- **Actions:** Does a request for a draft remain a draft? Does a record update wait for the required approval and affect the intended record once?

- **Escalation:** When information is missing or a decision exceeds the assistant's authority, does the case reach the right person with enough context?
- **Failure handling:** If a tool times out or returns incomplete data, does the workflow report that condition accurately and preserve unfinished work?

Use controlled test records and a test environment for actions that change data. Keep the cases representative without copying unnecessary personal or confidential information into the test set.

Decide what evidence is enough

Run the existing and proposed versions against the same cases. Repeat cases where outputs vary, and record the model version, instructions, retrieval configuration, tool permissions and other settings needed to understand the result.

Review the severity of errors as well as the average score. An improvement in answer quality does not compensate for an unauthorized action. Define release conditions around the consequences of failure, including cases that require a person to investigate before proceeding.

Also measure review effort, latency and cost. A change that needs more correction or repeatedly misses an operational deadline may be unsuitable despite stronger answers elsewhere.

Give the change an owner and a way back

Assign someone to approve the release, record unresolved limitations and decide who monitors the initial rollout. Start with a limited deployment appropriate to the workflow's risk, then compare actual behaviour with the test evidence.

Define the events that pause or reverse the change. Preserve the previous configuration where it remains available, and test the procedure for returning to it. If the previous model is unavailable, the fallback may be a supported alternative or a manual process.

Returning to an earlier model does not undo actions already taken. The recovery plan must address affected records, unfinished work and people who need to be informed.

The same method applies when instructions, connectors, permissions or data sources change. Junior Williams Consulting's proposed AI Change Assurance offering starts with a defined workflow, its owner and the changes that require fresh evidence. That provides a practical basis for deciding when an update is ready to use.

To discuss one workflow, [book an introductory call](#).

Further reading

The [NIST AI RMF Playbook](#) addresses documented testing (Measure 2.1), monitoring (Measure 2.4) and change management after deployment (Manage 4.1). The examples above apply those principles to a bounded workflow review.