WORKED EXAMPLE 01 · STATIC PRINT EDITION

Your AI gave a great answer. Why is the customer still doing the work?

From recommendation to action: checking that AI changes the right thing without breaking the rest.



Mike Idengren



Synopsis

A useful AI answer is only part of the job when a product promises to change a process. In this JarviSIM example, the user asks, “Make this process lower risk.” The chatbot recommends a specific control; the user chooses it and applies the change to the model. The reported structural check is one of one requested control added and seven of seven existing tasks retained. The three-run count is owner-supplied, not independently established by the screenshots. Whether the recommendation itself was good is a separate semantic evaluation.

A useful design principle spans all three priorities: reserve language-model judgment for interpreting and planning changes, and use deterministic functions wherever the operation can be specified precisely. This avoids unnecessary inference charges and waiting while making execution more predictable—benefits for cost, speed, and quality through reliability, rather than measured savings in this example.

Audience: AI Solution Product Managers · CIOs and CTOs · AI Developers

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

COST · SPEED · QUALITY

The business story

COST — Optimization priority. Use the lowest-cost model that meets the product's quality bar.

SPEED — Optimization priority. Reduce the time to a useful answer and completed change.

QUALITY — A process upgrade, not just a chatbot answer. The example asks whether the requested control was added, existing tasks were retained, and the structural result repeated.

Quality dimension	Result
Structural	**PASS** — 1/1 requested control added; 7/7 existing tasks retained
Stability	**PASS · owner-reported** — structural result reported across three runs
Semantic	**NOT ASSESSED** — recommendation quality needs its own rubric

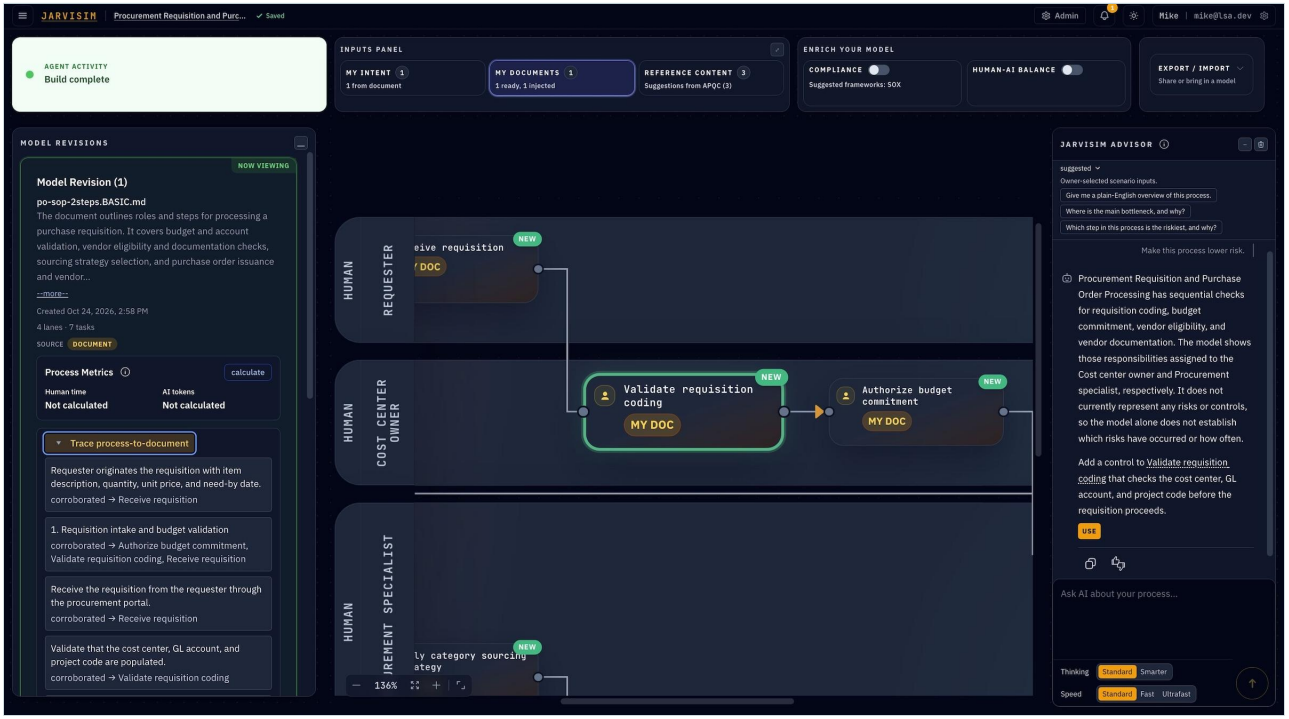
But a good recommendation is not the same as a completed action. Imagine an airline assistant finding a better connection and explaining why it fits. If you choose that option, but the booking never changes, you still have to finish the job. **But your booking never changes.** That gap—between advice and an outcome—is where a buyer should look closely.

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

JARVISIM COMPLIANCE QUALITY · PART I

The request is broad. The recommendation is specific.

The user asks, “Make this process lower risk.” The chatbot describes the purchasing process, notes that the model alone cannot establish which risks occurred or how often, and recommends a requisition-coding completeness control for the existing “Validate requisition coding” task.



Capture 1. From request to recommendation. Full supplied screenshot, uncropped.

The user’s request

Intent · “Make this process lower risk”

The broad request starts the journey. It is an expression of intent, not evidence that risk has fallen.

The chatbot’s explanation

Semantic · Not assessed

The explanation describes the process and its limits. Gray means recommendation quality has not been evaluated in this example.

The recommendation + Use

Structural · Starting point

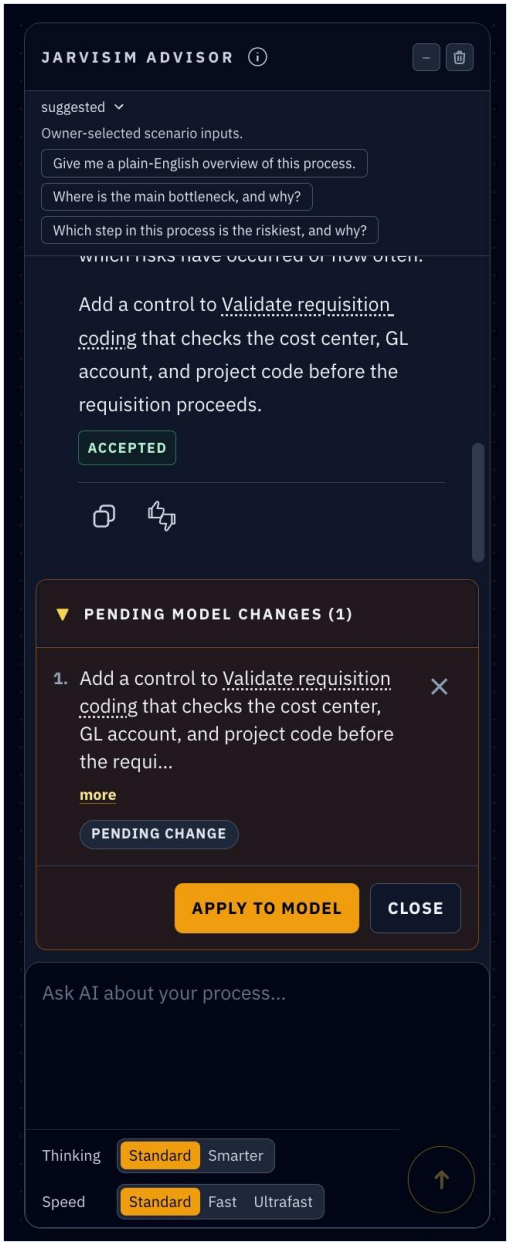
The specific proposal and Use button start the action path. Use selects a recommendation; it does not establish a successful change or stability across runs.

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

JARVISIM COMPLIANCE QUALITY · PART II

The user chooses. Tools carry out the change.

The user selects **Use**; the recommendation appears in **Pending Model Changes**. The separate **Apply to Model** action is the user's choice to proceed. The product then invokes its chatbot model-mutation operation (`apply_chatbot_mutation`) to make a candidate model change; the accepted revision is what the product checks next. This is an operation an AI product invokes—not just text for a person to copy into the process.



Capture 2. Selection is not application. Full supplied screenshot, uncropped.

Pending Model Changes

Structural · Selected, not applied

The selected recommendation is queued as one pending change. The model has not yet been shown to change.

Apply to Model

Structural · User consent to proceed

This separate action applies the selected proposal. The product's apply_chatbot_mutation mechanism is described in the story; this screenshot alone does not prove a tool invocation.

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

PART II · DIVISION OF LABOR

The LLM specifies the change. The tool executes it.

The LLM works out what should change. The heavy lifting happens while developing the recommendation: interpreting the request, considering what is already in the change queue, and working out how the Business Process Model and Notation (BPMN) model should change. The useful output is more than advice—it is a specification for the mutation.

> **Don't over-use the LLM for what should be reliable, fast, free deterministic functions.**

The tool carries out the specified change. The division of labor is simple: the LLM specifies the change; the mutation tool executes it. Deterministic model-editing functions apply precise instructions rather than asking the LLM to reason through the same decision again. That makes execution reliable, fast, and free of another LLM call. The product then checks the resulting model: was the requested control added, and were the existing tasks retained?

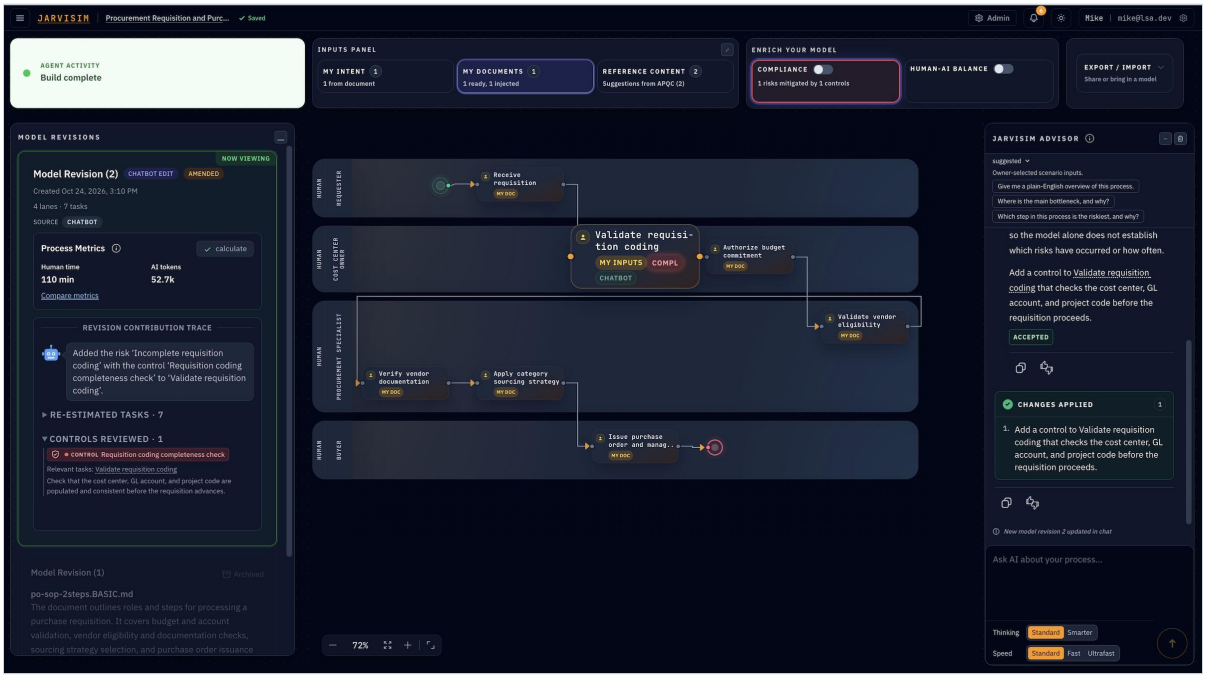
The design lesson. Use the LLM for judgment and planning. Use deterministic tools for execution. Structural quality checks whether the specified change was carried out correctly; stability checks whether that outcome repeats.

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

JARVISIM COMPLIANCE QUALITY · PART III

One control added. Seven tasks kept.

The revision view shows one control reviewed and seven tasks. Its contribution summary describes the requisition-coding completeness check. The selected “Validate requisition coding” task carries both **CHATBOT** and **COMPLIANCE** contribution labels. The control checks that cost center, general-ledger account, and project code are populated and consistent before the requisition proceeds.



Capture 3. Inspect the resulting revision. Full supplied screenshot, uncropped.

The robot’s revision summary

Structural · Reported change

The contribution trace says a requisition-coding completeness check was added to Validate requisition coding. This is the interface’s summary, not an independent tool-call trace.

The seven-task model

Structural · Retained process context

Inspect the seven task nodes in the resulting canvas. Counting visible tasks is not proof that all seven original task identities were retained; that comparison is a supplied outcome claim.

Re-estimated 7 · Controls reviewed 1

Structural · Visible counts

The revision labels RE-ESTIMATED TASKS · 7 and CONTROLS REVIEWED · 1. Re-estimation is not measured chatbot latency or a speed improvement; reviewed is not proof of control effectiveness.

Changes applied

Structural · Visible application receipt

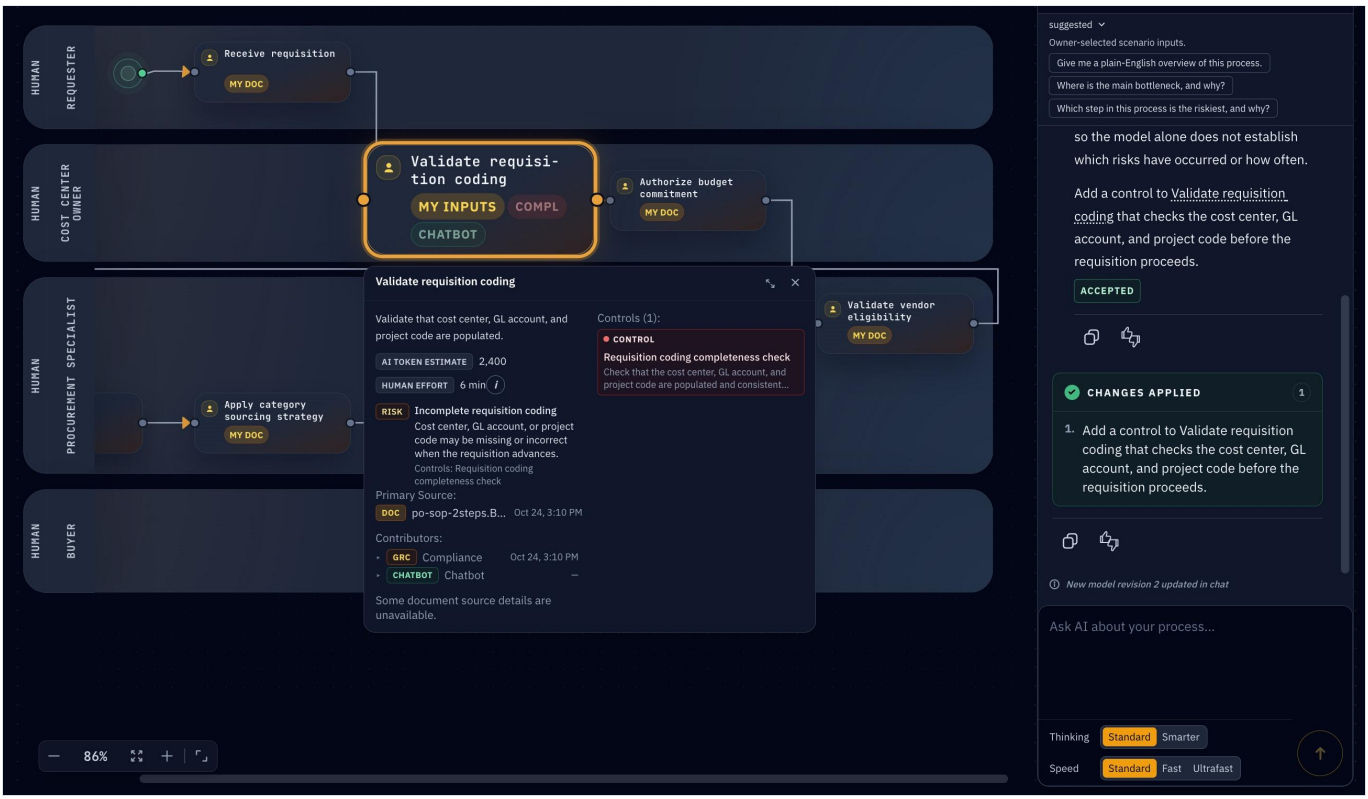
The advisor displays the applied-change message for this recommendation. One screenshot is not evidence of three separate runs or repeated stability.

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

JARVISIM COMPLIANCE QUALITY · PART IV

Inspect the control on the task.

Open the highlighted task details to see the attached control and the Chatbot and Compliance contributions.



Capture 4. Inspect the selected task. Full supplied screenshot, uncropped.

Validate requisition coding

Structural · The affected task

The selected node identifies where the change landed. The attached detail below lets the reader inspect the control rather than infer completion from chat alone.

The attached control

Structural · One visible control

Controls (1) identifies the requisition-coding completeness check. Its presence is a structural observation, not an evaluation of whether it reduces operational risk.

Compliance + Chatbot contributors

Structural · Displayed attribution

The task detail lists Compliance and Chatbot contributors. These interface labels describe attribution; they do not independently prove the underlying tool calls.

© 2026 Lean Scaled Architects LLC | https://lsadigital.com

RESULTS AND EVIDENCE LIMITS

A successful example is a start. Repeatability is another question.

The reported structural check found the requested control added (1/1) and existing task identities retained (7/7), with the outcome reported across three runs. That is a useful repeatability question for a buyer to ask, but the run count comes from the owner; the four screenshots do not independently establish it. The screenshots also do not prove task-identity matching on their own.

The revision panel labels the process-metric count **RE-ESTIMATED TASKS · 7**. Those tasks were re-estimated under the test conditions; this is not measured chatbot latency, customer productivity, or a speed improvement. A structurally correct change does not prove risk fell, the control works in production, or the recommendation was semantically good. Advice quality and model outcome need separate checks.

About this example and its evidence

The four historical captures show the request and recommendation, the pending/apply action, the applied revision, and the selected task detail. They are interface illustrations—not a standalone test report. The 1/1 control, 7/7 task-identity, and three-run figures are supplied outcome claims; no underlying run-by-run records or task-identity comparison artifacts are presented here. The screenshots alone cannot independently verify those claims. The three-run basis is owner-supplied, not inferred from the four images.

This is development evidence, not a customer deployment or proof of production effectiveness. It makes no claim of risk reduction, cost savings, latency improvement, customer acceptance, or compliance effectiveness.

For Claude Certified Architect–Professional readers

Anthropic's guidance on defining success criteria and building evaluations calls for criteria tied to the application's purpose and user needs. Here, the structural checks are distinct from semantic response quality: add the requested control, retain the seven existing task identities, and ask whether the outcome repeats. This is an application of that evaluation principle, not an Anthropic-prescribed JarviSIM workflow. The recommendation rubric and any judgment of operational risk are outside this article's evidence.

Source: https://platform.claude.com/docs/en/test-and-evaluate/develop-tests

© 2026 Lean Scaled Architects LLC | https://lsadigital.com