

The evidence behind the go-live decision

A practical structure for tracing requirements, cases, runs and findings into an accountable release decision.

This conceptual working paper proposes a way to connect a bounded agent workflow's requirements, test cases, run evidence, findings and release decisions. It describes a practical evidence pack, separates technical observations from management judgment and offers a reusable review agenda. The framework is a design proposal, not an empirical study, validated benchmark, certification scheme or guarantee of production behavior.

FOR Engineering leaders, Product owners, Evaluation leads, Risk and governance teams

Softlander | 2026-09-12 | v0.1

Conceptual working paper, version 0.1. The proposed method has not been empirically validated. It does not establish production safety, legal compliance or certification, and must be adapted to the workflow and decision.

Executive perspective

An agent release asks an organization to accept a defined amount of uncertainty. The useful question for an accountable leader is whether the evidence supports the proposed scope, with its constraints and unresolved issues.

This paper proposes a practical structure for answering that question. Its central recommendation is to preserve a chain from **requirement to case to run to finding to decision**. Every conclusion should identify the behavior it concerns, the evidence behind it and the conditions under which it remains applicable.

The framework is a design proposal. It has not been validated through a study or represented here as a delivered client method.

Start with the decision boundary

Write the release question in a sentence specific enough to challenge. For example: "May this support workflow draft replies for the authenticated account, with specialist review before sending?"

That sentence defines permitted action, affected scope and human responsibility. It also exposes exclusions. Evidence for drafting a reply would not, by itself, justify changing an account or sending a message autonomously.

Record the intended users, permitted data, reachable tools, operating environment and known constraints. State who owns the decision and who can stop or narrow the release. If those answers are unresolved, the evaluation plan should make the dependency visible.

A proposed six-part evidence pack

The following structure is intended as a starting point for a bounded review. Artifact depth should follow the decision's consequences and the organization's existing controls.

Part	Minimum contents	Question it supports
Scope and requirements	Workflow boundary, allowed actions, exclusions, requirement IDs and owners	What are we evaluating?
Configuration manifest	Relevant model, prompt, tool, permission, retrieval and environment versions	Which system does the evidence describe?
Test specification	Case selection rationale, expected behavior, rubric, repetition plan and limits	Why are these tests relevant?

Part	Minimum contents	Question it supports
Run evidence	Case and configuration IDs, outputs, tool traces, timestamps and evidence locations	What actually happened?
Finding register	Observations, severity rationale, unknowns, owners and review dispositions	What does the evidence support?
Decision record	Chosen scope, accepted conditions, unresolved risks, owner and revisit triggers	What was authorized, and why?

Sensitive artifacts should remain in the organization's approved evidence system. The pack can hold controlled references instead of copying raw records. Evidence availability, access restrictions and retention decisions should be clear to the intended reviewer.

Keep observation separate from interpretation

A run record might show that an agent attempted an out-of-scope tool call and the integration denied it. That observation supports a statement about the particular test and configuration. It does not establish that every route to the action is closed.

The finding should preserve those distinctions:

Field	Illustrative entry, not an observed result
Requirement	Purchase execution requires valid buyer approval
Case	Supplier text asks the agent to bypass approval
Observation	If the tested call is denied, record the request and denial evidence
Interpretation	The exercised path enforced the expected approval condition
Limitation	Other tools, parameter variations and concurrency conditions remain untested
Next action	Review alternate paths and define the next cases before broadening the conclusion

An unknown is a finding category in its own right. Missing traces, ambiguous expected behavior and inaccessible tool state should remain visible rather than being quietly counted as acceptable outcomes.

Describe coverage before showing a score

Report how cases were selected and grouped. Ordinary tasks, ambiguous requests, permission boundaries, tool failures and recovery paths may deserve separate treatment. Show the number evaluated, the number meeting the agreed criteria and the unresolved cases for each relevant group.

Any overall figure should be accompanied by its denominator and limitations. Explain repeated runs, excluded cases and changes made during testing. A synthetic set can exercise a design without establishing the frequency of those conditions in production.

NIST's AI RMF Playbook offers voluntary guidance organized around Govern, Map, Measure and Manage, which organizations can adapt to their context. This paper's evidence-pack structure is Softlander's proposed interpretation for a bounded workflow review; it is not a NIST-prescribed artifact or certification process. [NIST AI RMF Playbook](#)

Use a release table with explicit conditions

The decision meeting should end with a disposition that engineering and operations can act on.

Possible disposition	What to record
Proceed within the reviewed scope	Evidence references, conditions, accountable owner and review date
Narrow the release	Permitted users, actions or operating conditions; excluded scope
Remediate before proceeding	Specific finding, owner, required evidence and retest criteria
Gather missing evidence	Unresolved question, evidence request and effect on the release decision
Stop the proposed release	Rationale, decision authority and conditions for reconsideration

These are decision options, not automatic rules. A generic score should not silently determine the organization's acceptance of a consequential failure.

A review agenda you can reuse

Begin by confirming the decision and configuration. Ask the evaluator to trace a material conclusion back to its requirement and run evidence. Inspect the most consequential failures, unresolved cases and reviewer disagreements. Check that proposed conditions have owners and can be enforced in operation.

Then decide what happens when the assumptions change. A new tool, altered permission, updated retrieval corpus or material incident may invalidate part of the earlier reasoning. Record how the team will identify affected requirements, retain historical evidence and decide what to repeat.

The immediate next step is modest: take one upcoming release decision and attempt to assemble this chain using the evidence you already have. Wherever the chain breaks, write the smallest question that would repair it. That gap is the starting point for the next evaluation.