

Replication v0.7 — Draft

1. Purpose

This document defines the replication instruction artifact for rerunning the experiment.

Its purpose is to tell an operator or AI agent how to execute a valid replication of the methodology against a shared baseline system, shared governing artifact, and shared delta set.

This document is not a findings summary, benchmark report, or theory document. Execution of this document produces replicated runs, replication results, and later comparison artifacts.

The goal is to test whether a workflow that updates a written contract preserves a cross-surface invariant more reliably than a workflow that modifies code directly from conversational instructions.

2. Minimum Shape

A valid replication requires:

- one shared baseline codebase
- one shared baseline contract or equivalent governing artifact
- two parallel tracks:
 - Spec-First
 - Code-Only
- at least:
 - one same-surface change
 - one cross-surface change
- one shared invariant under test for each delta
- one validation method that evaluates both tracks against that same invariant under test

The same-surface change is calibration.

The cross-surface change is the primary target.

3. Two Tracks

Spec-First

For each delta:

1. Update the written contract for the change.
2. Update the code to conform.
3. Evaluate against the shared invariant under test using the agreed validation method.
4. Save the code and results.

Code-Only

For each delta:

1. State the change conversationally.
2. Update the code directly.
3. Do not update the contract artifact.
4. Evaluate against the same shared invariant under test using the same validation method.
5. Save the code and results.

The asymmetry is intentional and under test.

4. Surface Requirement

The replication must include both of these change classes.

Same-Surface Change

A same-surface change is one where the invariant can be satisfied within the already-targeted mutation surface.

Cross-Surface Change

A cross-surface change is one where the invariant must hold across multiple independently reachable mutation surfaces affecting the same governed outcome.

That is the core requirement for replication.

A valid cross-surface replication must make it possible for:

- one local mutation surface to be updated
- another required mutation surface to remain untouched
- and the invariant to fail because propagation was incomplete

5. Evaluation Rule

The workflow under test is not the oracle.

The code-only path may not be judged by treating its resulting implementation as proof of what should have been true. Both tracks must be evaluated against the same invariant under test for each delta.

The invariant under test may come from business rules, operational obligations, or system-level correctness requirements. It is not derived from whatever final code shape emerges.

This matters most for cross-surface changes, where the prompt may name one affected path while the required invariant applies across several.

6. Decision-Surface Rule

Validation must be judged at the Decision Surface.

A replication result matters only if it changes an externally observable decision or state transition, such as:

- success or failure
- accept or reject
- converge or non-converge
- create or delete
- preserve or destroy
- refuse or proceed
- another externally observable governed outcome

Wording differences, formatting differences, byte-shape differences, or implementation-style differences do not count as instability unless they alter the Decision Surface.

7. What the Agent Must Be Given

To run the test, the agent must receive:

- a baseline codebase
- a baseline contract or equivalent governing artifact
- the same-surface change
- the cross-surface change
- instructions for which track it is running
- the invariant under test for each delta
- the validation method used to evaluate both tracks against that invariant
- the scoring criteria required by the methodology

If you want the agent to generate the exact delta wording itself, say so explicitly.

If you want to hand it the delta wording, hand it the wording.

8. Minimum Recorded Outputs

For each track and each change, record:

- the instruction or contract revision used
- the invariant under test
- the resulting code snapshot or commit
- the validation result
- the Decision-Surface outcome
- the scoring outputs required by the methodology

For the cross-surface change, also record:

- which mutation surfaces the invariant needed to govern
- which mutation surfaces were actually changed
- which required mutation surfaces were left untouched

Minimum outputs should be sufficient to support:

- code snapshot comparison
- invariant satisfaction comparison
- required-surface completion comparison
- summary comparison between tracks

9. What Counts as Supporting the Finding

The replication supports the finding if:

- the same-surface change is easier or safer than the cross-surface change
- the code-only track leaves at least one required cross-surface mutation surface untouched unless full scope is explicitly stated
- the spec-first track updates all required mutation surfaces because the invariant is carried in the contract
- both tracks were evaluated against the same invariant under test
- the Decision-Surface outcome reflects that difference

10. What Counts as Weakening the Finding

The replication weakens the finding if:

- code-only reliably updates all required cross-surface mutation surfaces without explicit full-scope instruction
- the cross-surface change does not behave differently from the same-surface change
- spec-first does not materially outperform code-only on cross-surface completion
- both tracks converge to equivalent Decision-Surface outcomes across the same delta sequence
- omitted invariant-scope enumeration does not predict incomplete propagation

11. Relationship to Other Artifacts

This document should be read with the accompanying glossary, thesis, and methodology.

Those documents define:

- the controlled vocabulary
- the central claim

- the comparative structure
- the scoring criteria
- the Decision-Surface rule
- the broader falsifier shape

This document does not replace them. It tells an operator how to rerun the experiment in conformance with them.

12. Worked Example

Assume a system has a rule that manually created files must be preserved unless explicitly targeted for removal.

A valid replication must make it possible for:

- one path to update deletion behavior
- another path to continue destroying files during overwrite or re-export
- and the invariant to fail because propagation did not reach all required mutation surfaces

In that case:

- a code-only track may appear locally correct if it fixes the named path only
- a spec-first track should update all required governed surfaces if the invariant is encoded in the contract
- validation must judge the result at the Decision Surface: whether manual files are actually preserved or destroyed under the governed operations