Research Thesis v1.1 — Draft

1. Executive Thesis

AI reduces implementation cost.

What remains under-tested is whether it preserves stability under iterative change.

This work advances the claim:

In iterative development, workflows that externalize authority into a versioned contract preserve invariants more reliably than workflows that modify code directly through conversational prompts.

This is not a benchmark of one-shot coding ability. It is a stability-under-iteration thesis.

2. Core Comparison

This work compares two modification workflows, not two competing sources of truth.

Track A — Spec-First

- A requirement change is encoded in a versioned contract patch.
- Implementation is then revised to conform.
- The written yardstick evolves explicitly as the contract evolves.

Track B — Code-Only

- A requirement change is described conversationally.
- Code is revised directly.
- No contract artifact is updated.

The asymmetry is intentional and under test. One workflow carries forward a durable written statement of what must continue to hold. The other does not.

The code-only workflow is therefore not judged by treating its final implementation as its own oracle. It is judged by whether the invariant under test still holds.

3. What Is Being Measured

The question is not whether a model can satisfy a requirement once.

The question is whether earlier invariants survive later tightening, especially when a later change must hold across more than one mutation surface.

Evaluation is made at the Decision Surface: whether later changes preserve or violate the required externally observable outcomes and state transitions. Wording similarity, formatting similarity, and local code plausibility are not the primary measure.

4. Topology Boundary

The key distinction is between same-surface and cross-surface change.

A same-surface change tightens behavior within the surface already being modified.

A cross-surface change requires an invariant to hold across more than one mutation surface.

Earlier work established that same-surface tightening can remain stable or at least prompt-sensitive. Later work showed the more important structural boundary: cross-surface invariants fail predictably when full invariant scope is not explicitly enumerated.

This is the main boundary under study.

5. Immediate Mechanism

The immediate mechanism is explicit invariant-scope enumeration.

A sufficiently explicit prompt can succeed. But prompts naturally bind to the surfaces they name. Contract revision forces enumeration of all affected mutation surfaces.

So the issue is not simply prompt quality. The issue is whether required scope must be re-inferred during later modification, or remains durably encoded across revisions.

6. Minimal Experimental Shape

The minimal shape is:

- start from the same baseline system
- apply the same tightening sequence in two tracks
- let only the spec-first track update the contract artifact
- compare whether earlier invariants still hold after later changes

This becomes most informative when the sequence includes cross-surface changes, because those changes reveal whether the workflow preserved full invariant scope or only the locally named surface.

7. Falsifiable Claim

This thesis is falsifiable.

It weakens if code-only workflows exhibit invariant preservation, regression behavior, and collateral change equivalent to spec-first workflows across the same tightening sequence.

It also weakens if cross-surface changes do not show greater instability than same-surface changes, or if omitted invariant-scope enumeration does not predict incomplete propagation.

8. What This Is Not

This work is not:

- a coding benchmark
- a model ranking exercise
- a test of raw model intelligence
- a claim that AI cannot code

Its narrower claim is that where invariants are maintained changes how reliably they survive later modification.