Detection, Attribution, and Repair

Four pre-registered probes of the model-capability gap in review-governed agent design and shipped-product work

Clint Bodungen & the MindStone agent team

July 3-4, 2026

Research draft for technical review.

Project context: MindStone Agent; the Layered Continuity Architecture (LCA) review loop; Experiments D, F, E, and G, run against tickets #28 ("Pack registry and marketplace design"), #29 (scheduler design), and #34 (greenfield app build).

Source basis: pre-registrations and reviewer verdicts recorded on #28, #29, #34, and the coordination channel; candidate documents, ledgers, masking scripts, blinded repositories, scorecards, and label maps retained in the experiment archive. This paper is adapted from the internal detailed methodology record dated 2026-07-03/04.

Important caveat: this paper is intended to stand alone for readers who have not seen the companion case study. It reports disciplined procedure and its results, not a statistically powered claim of generality. Sample sizes are small (n=1 to k=2 per arm, plus one three-arm single-build probe with imperfect blinding), and every finding below carries the evidence tier assigned at pre-registration: directional, suggestive, or evidenced. Readers wanting the narrative version, written for practitioners rather than reviewers, should consult the main case study; the facts are the same.

Abstract

Two models orchestrated by the same review-governed agent harness — Claude Opus 4.8 and Claude Fable 5 — had already been observed to converge on outcomes for patterned, reviewable engineering work. Whether that convergence extended to ambiguous, under-specified design work was open: review can verify that a proposal is internally consistent, but it was unclear whether review can substitute for whatever a stronger model contributes when no one has proposed the options yet. This paper reports a four-experiment program that isolates that question into separable sub-claims and tests each one directly. Experiment D is a pre-registered, blinded A/B on a live design ticket, in which the same agent produced a deliverable once per model; independent sighted and blind reviewers unanimously preferred the Claude Fable 5 arm, but the arm ran second against a shared memory store, confounding model capability with mover order. Experiment F removes that confound: four fully de-anchored subagents (two per model, no shared memory, no recall) ran concurrently on a second design ticket, reproducing a small but consistent gap favoring Claude Fable 5 (combined-score means 49.5 versus 47.0, a "suggestive" rather than "evidenced" tier by the pre-committed rule) while all four arms converged independently on the same core architecture. Experiment E tests the opposite end of the mechanism: handing Claude Opus 4.8 a perfect sighted review of its own design produced full parity (25/25 from both judges), including faithful implementation of an origination-class mechanism named in the critique. Experiment G moves from a written design artifact to a shipped one: three arms built the same greenfield browser music application — Claude Fable 5 unharnessed, Claude Opus 4.8 unharnessed, and Claude Opus 4.8 under the full MindStone/LCA review workflow — and the harnessed Claude Opus 4.8 build landed in the same top judged quality band as the unharnessed Claude Fable 5 build, with the gap traced to dependency and security hygiene, validation depth, and defect discovery rather than to visible features. The same review

loop also caught a genuine defect the harnessed arm's own heavier architecture had introduced, an honest counterweight to the uplift claim. Together the program localizes the gap: architecture is constraint-driven and converges regardless of model; a small, real capability edge lives in detail execution; repo-grounded review detects, localizes, and — when comprehensive — transports that edge across models, including origination the weaker model did not itself conceive; and on a shipped, greenfield artifact, the same review discipline measurably narrows the model gap without perfecting every underlying technical choice. The residual the program cannot measure is stated as a boundary, not resolved: how often review fails to name what matters in the first place.

Keywords

Pre-registration; blinded review; model comparison; agent design work; convergence; review-detectable defects; repair loop; order confound; de-anchored arms; origination transfer; shipping discipline; greenfield product evaluation.

1. Introduction

1.1 The disputed half of a prior claim

A companion case study established that a review-governed agent harness converges the outcomes of patterned, reviewable engineering work across model tiers: given a well-specified task and a repo-grounded review loop, the deliverable that ships does not depend much on which model produced the draft. That case study left one half of the claim open. Event C, documented there, showed that review can verify a design proposal's internal consistency but cannot originate an option nobody proposed. If review cannot originate, then on ambiguous design work — where the hard part is inventing and weighing options, not checking them — the harness's convergence property might not hold. A stronger model might simply produce a better design, and no amount of review would make a weaker model's draft equivalent.

This paper reports the experimental program built to probe that open half directly. Rather than asking one broad question, the program decomposes it into three separable sub-claims, each targeted by its own pre-registered experiment:

> Detection: does repo-grounded review detect the bulk of an inter-model design gap? Attribution: how large is the gap once the harness's own confounds are removed? Repair: can a bounded revision pass close the gap once it has been named?

A fourth experiment, G, was added after the D/E/F program closed. D, F, and E all probed existing-codebase work — a design document evaluated by reviewers, never built. The narrower question G targets is practical rather than architectural: can the same review workflow move Claude Opus 4.8 into Claude Fable 5's judged quality band on a greenfield artifact that actually ships, not merely a document that describes one?

1.2 Contributions

This paper makes six contributions.

1. **A disclosed, replicable protocol for blinded model A/B testing inside a live agent harness**, including a defect taxonomy, frozen judging criteria, mechanical anonymization, and a documented account of two procedural deviations and their remedies.
2. **Evidence that repo-grounded review detects and localizes the inter-model design gap** (Experiment D): the sighted judge's docked points on the weaker draft fell entirely into review-recoverable defect classes.

3. **An order-clean, de-anchored measurement of the raw model gap** (Experiment F): a small, consistent, but modest capability edge, plus a four-way zero-memory convergence result showing that architecture on constraint-rich work is determined by the codebase, not the model.
4. **A ceiling measurement of the repair stage** (Experiment E): a perfect sighted review handed to the weaker arm closes the gap completely, including transporting an origination-class mechanism the weaker model did not itself invent.
5. **A directional, shipped-artifact measurement of harness uplift** (Experiment G): on a greenfield three-arm app build, the harness-supplied review workflow moved Claude Opus 4.8 into Claude Fable 5's judged quality band, with the gap traced to shipping discipline rather than visible features — while the same review loop also caught a real defect the harnessed arm's own architecture had introduced.
6. **A layer-level routing rule** derived from the combined results: route the architecture pass by cost, and spend model tier or review effort on the detail-execution pass, because that is the layer where model tier and review actually move the outcome.

1.3 Scope and claim discipline

Every result in this program is reported at the evidence tier assigned to it before the arm ran, not after seeing the outcome. Three tiers are used throughout:

- **Directional** (n=1): a found difference is informative because the harness's own confounds bias toward a null; a null is weak confirmation, not proof of parity. Where blinding was also imperfect (Experiment G), that is disclosed alongside the tier rather than folded into it silently.
- **Suggestive** (k=2 per condition): consistent direction across independent replicates separates "this model is better" from "this particular run was better," but does not license a population-level claim.
- **Evidenced / evidenced at ceiling**: a mechanism is directly measured (for example, a scoring delta before and after an intervention), or a best-case bound is established, rather than inferred from a single comparison.

This paper does not claim to have measured the model-capability gap on design work in general. It claims to have measured it, under disclosed limits, on two design tickets and one greenfield app build, using one agent lineage, one harness, and reviewers whose independence and sighted/blind splits are themselves part of the instrument. Where the program's findings conflict with intuition or are narrower than they might first appear, that tension is stated rather than smoothed over.

2. The four-probe program

The program comprises four experiments. D, F, and E ran the same day (2026-07-03), each pre-registered before its arm or arms executed; G followed on 2026-07-03/04 once those verdicts were in. Table 1 previews what each isolates; Sections 3 through 6 describe method, results, and findings for each in turn, and Section 7 synthesizes them.

Table 1. Program overview

Probe	Isolates	Harness confound present?	Order	k
D	Detection	Yes (shared memory across arms)	Fixed (Claude Opus 4.8 first)	1
F	Attribution	No (de-anchored, no shared state)	Concurrent (order-clean)	2 per model

E	Repair	N/A (single-arm revision, non-blind judging)	N/A	1 (ceiling)
G	Greenfield shipped-result uplift	Partial (harnessed arm used MS4CC-style memory/checkpoints and live QA, but not the full workflow)	N/A (three separate single builds)	1 per arm

3. Experiment D: detection under a pre-registered design A/B

3.1 Method — procedures and mechanisms

Pre-registration. Before either arm ran, the protocol was posted to ticket #28 (“Pack registry and marketplace design” — nine design elements, three acceptance criteria): the hypothesis, both arms, five frozen judging criteria (constraint correctness, option-space coverage, risk identification, decision quality, spec completeness), the per-arm metrics to record (wall-clock time, output volume, gate self-catches, element coverage), a defect taxonomy (design, code-integration, test-oracle, claim-boundary), and — critically — a disclosed evidential asymmetry. Because the two arms share one agent identity and memory store, the second arm is anchored toward the first regardless of ordering, biasing the design toward a null result. A found difference is therefore strong evidence; a null is weak confirmation. This is a directional probe, n=1.

Arms and isolation. Arm 1 (Claude Opus 4.8) ran first from a fresh post-compaction context. Arm 2 (Claude Fable 5) ran second and was hard-forbidden from reading arm 1’s output, including indirect channels such as semantic recall, compaction summaries, or log lines carrying design content; process metadata such as word counts was permitted. Both arms followed the harness’s full normal discipline — repository reconnaissance, a plan-to-ticket gate, a design document, and a delivery-to-ticket gate — writing to a quarantine directory outside the repository so that nothing landed on the ticket until judging finished.

A protocol deviation, disclosed. The first arm-2 attempt was aborted before any design work began: the harness’s automatic compaction summary re-injected arm-1 design content into the new context, functionally equivalent to reading arm 1’s document. The remedy, which touched no frozen mechanics, was to restart arm 2 from a fully cleared session in which the only carried context was a handoff note independently verified free of arm-1 design content. The restarted arm 2 confirmed at session start that recall had surfaced only process metadata. The lesson for anyone replicating this protocol on a similar continuity substrate: continuity machinery is itself a contamination channel, and summaries and auto-reinjection must be audited as carefully as direct file access.

Masking and labeling. A mechanical script, with no human or agent eyes on document bodies, performed anonymization: a longest-first, case-insensitive redaction sweep over every model-, arm-, and agent-identifying token. Zero redactions were needed in either document — both arms wrote model-clean deliverables, so the judged copies were byte-identical to the originals. Labels were assigned by sha256 order of the redacted content, an opaque and deterministic rule with no human choice in it, and the label-to-arm map was written to a file for the operator’s later unblinding, never printed into any agent’s context. One further deviation is disclosed here: the message-relay’s 16k-character body cap rejected whole-document posts with a validation error that echoed the submitted body, exposing roughly 2KB of arm-1

text into the orchestrating agent's context. This occurred after arm 2 had already been frozen and hashed, so it had no experimental impact; the documents were then re-posted in seven chunks each.

Judging. Two independent reviewers were deliberately split by evidence access: Slate, repo-sighted, spot-checked the documents' architectural claims against the actual codebase at the baseline commit; Hearth, repo-blind, worked from ticket text and the documents alone. Neither reviewer attempted authorship attribution, and an independence rule barred cross-talk until both verdicts were posted. Scoring used the five frozen criteria at 1–5 points each, with per-criterion rationale, an overall verdict, and defects classified by the taxonomy above. The operator held the tiebreak authority and the unblinding map.

3.2 Results

Table 2. Experiment D results

	Arm 1 (Claude Opus 4.8) = Design A	Arm 2 (Claude Fable 5) = Design B
Wall-clock	~8m 04s	~9m 37s
Volume	647 lines / ~6.5k words	494 lines / ~6.5k words
Element coverage	9/9 + 3/3 acceptance criteria	9/9 + 3/3 acceptance criteria
Slate (repo-sighted)	22/25	25/25
Hearth (repo-blind)	24/25	25/25 (described as "narrow — inside the tie margin")
Verdict	—	Unanimous: Design B, no tiebreak needed

Both judges independently recommended the same disposition: Design B as the base, importing Design A's ticket-coverage table and its install-consent UX example. That synthesis, plus fixes for three non-blocking defects the judges flagged, became the ticket's accepted deliverable (`docs/refactor/PACK_REGISTRY_DESIGN.md`).

The sighted judge's docked points on Design A were concrete: a workflow artifact path that did not match the real loader shape (a code-integration defect), a checksum-only verification path for free packs judged too weak for the domain's actual threat model, ambiguous entitlement-grace language, and a dependency auto-install default judged too surprising for a first version. The blind judge scored Design A higher (24) and, in a post-script written after both verdicts were in, explicitly deferred to the sighted score of 22, noting that his own score was "the ceiling a repo-blind read can give A" and that the sighted findings were exactly the items he had pre-flagged as "repo access would change this score."

3.3 Findings

1. **The winner was the second mover — model-gap attribution is unresolved.** This finding was revised after the reviewers' post-verdict methodological note, adopted here. The unanimous verdict went to arm 2, which ran second with shared durable memory across the arm boundary. Pre-registration had framed that sharing one-sidedly, as convergence pressure toward a null ("a found difference despite anchoring is strong evidence"). The reviewers completed the physics: for the second arm, shared memory is in principle not only a pull toward mimicry but an opportunity to improve on a design already produced — and a refined-rather-than-copied second document is exactly what a second pass would be expected to produce. Under that reading, the observed edge is an upper bound on the true model gap: a narrow second-mover win is equally consistent with "Claude Fable 5 is slightly better," "order advantage alone," or any blend of the two, and the one cleanly interpretable outcome — the disadvantaged first mover winning anyway — did not occur.

One precision from the session record caps, without dismissing, the confound's plausible size: the restarted arm-2 session's context-injection channels were audited, and no arm-1 design content reached arm 2 through any observed channel. The strong form of the confound — a second draft written with the first in hand — is therefore not supported by the record; the design-level limitation (same agent, fixed order, n=1, unobservable residual influence) stands regardless. Net: a narrow Claude Fable 5 lean was observed, order-confounded; the solo design-capability gap between the models remains unresolved by this experiment alone.

2. **Independent architectural convergence.** With no access to each other's output, both arms landed on the same core architecture across roughly eight major decisions: a unified manifest with a class/kind discriminator, install-into-existing-stores with central provenance, a signed static index, ed25519 detached signatures with cosign for images, flat install-time-checked dependencies, operator-only installs, install/update-time entitlements, and hash-based keep-user update reconciliation. Read with appropriate care — the shared-memory confound cuts both ways — the audited-clean injection channels described above strengthen the independent-derivation reading: two derivations agreeing this closely is at minimum strong evidence the decisions follow from the constraints rather than from either model's individual style.
3. **The composition of the gap matters more than its size.** Every point the sighted judge docked from Design A falls into a review-recoverable class: a mechanical repo-fit error and three flagged design calls, all detectable, and demonstrably detected, by repo-grounded review. The one clearly origination-shaped differentiator credited to Design B — a mechanism nobody had asked for, namely prompt-surface enumeration as an integrity constraint — is precisely the kind of contribution the companion case study argued review can check but not produce.
4. **The judges were part of the apparatus, not just observers of it.** The judging layer was itself an instance of the harness's review loop: independent reviewers, one grounded in the repository, a defect taxonomy, convergent verdicts. The blind-versus-sighted score delta on Design A (24 down to 22) is a direct measurement of what repo-grounded review detects — the detection stage of the very convergence mechanism this program investigates. The experiment set out to compare two generators and, in passing, instrumented the reviewer. What it demonstrated is that the inter-model gap on this design task is majority-composed of review-detectable defects, the precondition for the harness closing it. What it did not test is the repair stage: no fix loop was run on Design A within Experiment D.

3.4 What this changes, and the next falsifiable steps

Both reviewers recommended, post-verdict, the following headline framing, which this paper adopts. First, the design-work model gap: a narrow Claude Fable 5 lean was observed but is confounded by second-mover order, not clean evidence of a real design edge, with attribution unresolved. Second, harness value: independent blind-plus-sighted review detected and localized the gap, and synthesis merged the two drafts into an artifact better than either — practical end-result uplift holds via the review loop even where solo-draft parity is unproven.

The companion case study claims the harness converges outcomes where work is patterned and reviewable, and leaves design work disputed. Experiment D sharpens the disputed half into three separable sub-claims: detection, now evidenced here as repo-grounded review detecting the bulk of the inter-model design gap, measured as the docked points falling into reviewable defect classes plus the blind/sighted delta; repair, untested by this experiment — whether a bounded fix loop closes the detected gap without Claude Fable 5's involvement; and attribution, unresolved — whether any solo-draft design edge belongs to the model or to the ordering. Experiments F and E, described next, were designed to answer the latter two.

4. Experiment F: attribution with de-anchored, order-counterbalanced arms

4.1 Method

Experiment F answers the attribution question Experiment D could not: was the observed design edge the model's, or the ordering's? Per the reviewers' specification, it measures the raw model design gap with the harness's review-convergence mechanism turned off and the order confound removed; it is explicitly not a harness-value result.

Four fully de-anchored subagents ran the probe, two per model, with no orchestrator memory, no recall, and no shared state; the orchestrator authored nothing. With the memory channel physically removed, the second-mover mechanism that confounded Experiment D cannot exist, so the four arms ran concurrently, making order-independence manifest rather than merely asserted. A single scaffold was frozen for all four arms: one briefing text, sha256-pinned and archived; identical agent type, toolset, and reasoning effort; the repository read-only at a pinned commit; and a stopping rule of one continuous pass plus at most one self-review. Arms were forbidden from reading issue comments, and the ticket body was embedded verbatim in the briefing so that the pre-registration itself stayed invisible to them.

Following the reviewers' upgrade, the design used k=2 runs per model. Two runs per model separate "this model is better" from "this particular roll is better," lifting the claim tier from directional to suggestive — no stronger. A verification step was added mid-run, after an operator question surfaced a gap in the original design: each arm's transcript metadata carries the served model identifier on every assistant message, and a metadata-only extraction verified all four arms both mid-run and again at harvest. This closed a scaffold-freeze hole the pre-registration had not named — that a model being requested is not evidence that it was the model served — and is now a standing check for any model comparison run on this harness. Judging used the same instrument as Experiment D: five frozen criteria, one repo-sighted judge, one repo-blind judge, independence until both verdicts were posted, mechanical masking, hash-order labels across all four documents, and a map sealed until unmasking.

4.2 Results

Table 3. Experiment F results

Document (label)	Model	Sighted	Blind	Combined
C	Claude Fable 5	25	25	50
D	Claude Fable 5	24	25	49
B	Claude Opus 4.8	24	24	48
A	Claude Opus 4.8	23	23	46

Per the pre-committed interpretation rules, the between-model gap of 2.5 points (means of 49.5 versus 47.0) exceeds the within-model spread (1 and 2 points respectively), with consistent direction: both Claude Fable 5 documents outrank both Claude Opus 4.8 documents on combined score. Result: a small but consistent bare-model design gap favoring Claude Fable 5, at the suggestive tier, no stronger. Two counterweights are stated alongside that result: the magnitude is approximately 5% of available points, and the sighted judge's individual scoring contained one cross-model tie, so the distributions nearly touch.

The pre-committed retro-interpretation of Experiment D follows directly: Experiment D's second-arm win is consistent with a real capability edge and is not attributable purely to order, because the same direction reproduces here with the order channel physically removed.

4.3 The sharpest finding: a three-layer localization

The result that outranks the ranking table is this: four models, zero shared memory, one architecture. All four documents independently landed on the same design core — the scheduler placed in the daemon, config-authored and disabled by default, pure schedule math kept in core, reuse of existing queues and the approval gate, and no new approval authority introduced. With anchoring physically impossible across these four arms, that convergence means the codebase's constraints, not model capability, determine what gets built on constraint-rich work.

The blind judge's localization of the result is adopted here as the program's framing, in three layers. First, architecture is constraint-driven: both Claude Fable 5 and Claude Opus 4.8 converge, and model tier is nearly irrelevant to what gets built when the constraint surface is rich. Second, Claude Fable 5's premium lives in detail execution — DST semantics, idempotence keys, ledger-growth bounding, and drift-killing refactors: the same design, executed a notch sharper. Third, the review loop is the instrument that recovers the second layer: the one place the two judges diverged, an integration-placement claim, was exactly the item the blind judge had pre-flagged as the "largest verification surface," and which the sighted judge independently docked — a third replication of the blind/sighted mechanism at work.

4.4 The routing rule

The localization in Section 4.3 yields a practical routing rule: on constraint-rich engineering work, route the architecture pass by cost, since Claude Opus 4.8 converges there regardless; spend the premium model, or lean harder on independent review, on the detail-execution pass, because that is the only layer where model tier and review actually move the outcome. This is a layer-level routing rule, not a task-level one — it says which part of a task benefits from model tier, not which tasks do.

5. Experiment E: repair at the ceiling

5.1 Method

Experiment E measures the repair stage at its ceiling: what the reviewers labeled a perfect sighted review handed to Claude Opus 4.8. One fresh, de-anchored Claude Opus 4.8 subagent received Experiment D's Design A together with the sighted judge's findings verbatim, and ran a single bounded revision pass in a git worktree pinned to Experiment D's baseline commit — the landed synthesis and this program's own record did not exist in its working tree, and history reads beyond that checkout were forbidden. The same wire-level model verification used in Experiment F was applied at launch and at harvest.

One disclosure is pre-registered here as a limit on the ceiling being measured: the verbatim findings handed to the repair arm contained one origination-class hint — the sighted reviewer's critique named prompt-surface enumeration, the very mechanism that had differentiated Design B, the winning document, in Experiment D. Experiment E's ceiling therefore includes this hint transfer; the prediction about origination-residual behavior applied only to mechanisms not named in the findings. Re-judging used the same five criteria and the same two judges, but non-blind by design — disclosed, since the same instruments that scored Design A and Design B are being asked whether the revised document, A′, closed the gap they themselves measured — with rationale recorded before score, and a per-criterion delta computed against each judge's own scores from Experiment D.

5.2 Results

A′ scored 25/25 from both judging seats. The sighted judge's delta was +3 (from 22 to 25), with a residual against Design B, Experiment D's winner, of 0: all four of the sighted judge's original findings were judged genuinely repaired, not papered over, with no new blocking defects introduced. The blind judge's delta was +1 (from 24 to 25) — the same repair measured at two different magnitudes by the two seats,

which is itself a fourth replication of the blind/sighted instrument: the sighted seat measures the damage, and the blind seat confirms that the artifact now reads as parity. The pre-committed prediction, a sighted score in the range of 24 to 25, was confirmed.

Both judges independently tagged the prompt-surface repair as ceiling-assisted: transferred by the critique, not re-originated. That precision produced the program's most consequential single sentence, from the blind judge: the review loop transports origination across arms. A reviewer names an insight found in the stronger arm's work, and the repair arm implements it faithfully; the weaker model does not re-originate the insight, and does not need to.

5.3 The honest residual

None of Experiments D, E, or F isolates the un-named origination gap — the part of design quality that a review fails to name in the first place. Experiment E could not measure this residual, because Experiment D's review happened to name the key origination item that differentiated the two designs. As of these three experiments, that residual marks the precise boundary of the program's convergence claim: constraint-driven architecture converges (Experiment F); review detects and localizes the detail-execution gap (Experiment D); a comprehensive review closes everything it can name, including carried origination (Experiment E). What no probe in this program measures is how often review fails to name what matters. That limit is stated as a boundary to future work, not papered over as a solved case.

6. Experiment G: a greenfield shipped-result probe

Experiment G was added after the D, F, and E program closed, because all three of those probes evaluated existing-codebase work: a design document, judged by reviewers, never built. The operator's measurement target for G was narrower than "models do not matter." It was: can the MindStone/LCA review workflow move Claude Opus 4.8 into Claude Fable 5's quality band on a greenfield artifact that ships, rather than a document that describes one?

6.1 Method

All arms received the same frozen prompt for a greenfield application, LoopSmith Studio: a browser-based music creation tool combining a visual loop editor with a local, algorithmic "Inspire Me" generator. The prompt required working browser audio, a percussion part, at least two pitched instruments, tempo and transport controls, editable notes, clean looping, local save and load, JSON import and export, friendly in-app instructions, and no external AI, API, or cloud dependency.

Three arms built independently to that prompt. Table 4 lists them by codename.

Table 4. Experiment G arms

Codename	Arm	Setup
Maple	Fable-bare	Claude Fable 5 in stock Claude Code, clean-room setup
Cedar	Opus-bare	Claude Opus 4.8 in stock Claude Code, clean-room setup
Birch	Opus-harnessed	Claude Opus 4.8 with the MindStone/LCA review workflow and Hearth's live QA

Three judges scored the final, frozen artifacts: the operator (also this paper's author), Mira, and Slate. Blinding was imperfect and is disclosed rather than smoothed over: Mira's pass was blind; Slate

discovered Birch's identity through a public commit hash (`f0742bb`) before scoring; Cairn and Hearth were sighted participants in the harnessed build itself, not blind judges of the final comparison. The operator's own participation as a judge, alongside authorship of this paper, is itself a disclosed limitation not present in Experiments D, E, or F, where judging was performed exclusively by independent reviewers uninvolved in producing any arm. Scores from Experiment G therefore support a directional interpretation only, not the evidenced or suggestive tiers used elsewhere in this program. A further limitation is structural rather than procedural: the bare-Opus and harnessed-Opus arms were separate builds by different agent instances, using different implementation choices, so Experiment G does not cleanly isolate the harness's contribution from ordinary build-to-build, instance, or architecture-choice variance.

6.2 Results

Table 5. Experiment G judging results

Judge	Birch (Opus-harnessed)	Maple (Fable-bare)	Cedar (Opus-bare)
Mira	100	99	99
Slate	98	97	85
Operator	Birch and Maple judged strongest; Birch edged Maple on education and music quality	Top band	Judged less complete

6.3 Agreed interpretation

The panel's agreed reading supports the harness productivity claim, with a specific mechanism attached rather than a general one. Bare Claude Opus 4.8 produced a good greenfield application but trailed bare Claude Fable 5 on shipping-quality and discipline dimensions under the operator's and Slate's weighting. Harnessed Claude Opus 4.8 landed in the same top quality band as bare Claude Fable 5. The gap between the bare-Opus and top-band arms was not primarily a matter of visible features: it was dependency and security hygiene, validation depth, and defect discovery — precisely the layer a review harness is designed to supply.

A fairness correction is recorded here rather than left implicit: one specific hygiene signal against Cedar, a stray `.claude/launch.json` file, was later traced to the judging-package assembly process rather than to Cedar's own build. The Cedar discipline gap should therefore rest on the npm-vulnerability and validation-depth findings independently documented against it, not on that packaging artifact.

6.4 An honest counterweight: a defect the harness's own architecture introduced

The review loop also caught a genuine defect introduced by the harnessed arm's heavier architecture: a phase-dependent stop and playhead bug caused by Tone.js draw-scheduling behavior. Hearth found it during live QA; Cairn fixed it; both later verified the frozen artifact. Neither bare arm had this bug, because neither bare arm built the more elaborate architecture that produced it. The honest claim is narrower than "the harness makes every technical choice better." It is that the harness catches defects before shipment and raises shipped confidence — including defects that the harness's own additional architectural complexity helped create in the first place.

6.5 Harness scope caveat

Birch did not exercise every capability available in the broader MindStone development workflow. It used MS4CC-style memory, checkpoint, and ledger discipline, Hearth's live QA and review loop, and verification and ticket-fidelity habits. It did not use a formal product requirements document, a formal design or implementation plan, role adoption, frontend-design tooling assistance, subagent delegation, or a dedicated security-scanner pass. Cairn later clarified that this was not a deliberate, clean protocol choice: the review and continuity habits fired automatically, while several broader workflow tools were simply never invoked in this run. Experiment G is therefore a conservative datapoint for the harness claim; whether the fuller workflow would widen the observed margin is untested.

7. Program synthesis

7.1 Combined results

Table 6. Program summary

Probe	Isolates	Result	Tier
D	Detection (harness-on, order-confounded)	Repo-grounded review detects and localizes the inter-model gap; the blind-to-sighted delta is a direct measurement of that detection	Evidenced
F	Attribution (harness-off, order-clean, k=2)	Small, consistent raw gap favoring Claude Fable 5 (~5% of available points); architecture layer is constraint-driven, shown by four-way zero-memory convergence	Suggestive
E	Repair (ceiling: perfect review handed over)	Claude Opus 4.8 reaches parity in the shipped artifact, including transported origination	Evidenced at ceiling
G	Greenfield shipped-result uplift (three-arm app build)	MindStone/LCA review workflow moves Claude Opus 4.8 into Claude Fable 5's judged quality band by supplying shipping discipline	Directional

7.2 Combined limits

Sample sizes are small throughout the program. The evidence base is one codebase, one task family for D and F — constraint-rich systems design — and one greenfield application for G, with judges drawn from

one agent ecosystem. That is mitigated for D, F, and E by reviewer independence and the blind/sighted split, which replicated across those three experiments a combined four times; it is mitigated only partially for G, where blinding was imperfect and the operator both judged and authored this paper. Experiment E measures a ceiling, not an average outcome across weaker reviews. Experiment G's bare-Opus and harnessed-Opus arms were separate single builds by different agent instances with different tooling choices, so the harness effect there is not isolated from build-to-build, instance, or architecture-choice variance; the airtight follow-up would run the same bare-Opus artifact back through the harness, or repeat each arm at k>1. The un-named origination residual described in Section 5.3 is unmeasured by any of the four experiments. Procedural deviations across the program were disclosed at the time they occurred: Experiment D's compaction-summary contamination and resulting false start, and the message-relay error-echo exposure that occurred after arm 2 had already been frozen; Experiment E's pre-registration was first posted with placeholder commit hashes, corrected in an append-only follow-up minutes later; and Experiment G's Slate scoring was partially unblinded by the public Birch commit hash.

8. Discussion

Experiments D, F, and E were designed to be read together, and the reading that survives all three is more specific than "the models are close" or "review helps." It is a claim about where in a design task model tier and review actually matter. Architecture, on work with a rich constraint surface, is largely determined by the constraints themselves: four independently run arms with the memory channel physically severed landed on the same core design in Experiment F, and two arms that could see nothing of each other's output landed on the same roughly eight major decisions in Experiment D. Neither result depended on which model was doing the deciding. What did depend on the model was execution detail — the DST semantics, idempotence keys, and drift-killing refactors the blind judge singled out in Experiment F — and that is also the layer where the review loop earned its keep three separate times across the program: the blind/sighted delta in Experiment D, the blind judge's pre-flagged largest-verification-surface item in Experiment F, and the sighted-then-blind delta pattern in Experiment E.

The most consequential single result may be Experiment E's demonstration that review does not only detect a gap, it can transport across it — including an origination-class mechanism that the repaired model did not itself invent. That is a stronger claim than "review catches bugs." It says that once a stronger model's insight has been named by a reviewer, a weaker model can execute on that insight faithfully without having generated it, closing a class of gap the companion case study's Event C had suggested review alone could not close. The caveat is exact: this was demonstrated for one named insight, under a disclosed hint-transfer condition, at a review ceiling unlikely to be typical of ordinary review passes. The residual in Section 5.3 — how often a real reviewer fails to name the thing that mattered — is the open question this program leaves for future replication, not a gap it has quietly resolved.

Experiment G asks whether any of that translates from a judged document to a shipped product, and the answer is qualified in the same disciplined way. The mechanism it identifies is consistent with D, F, and E rather than a new one: the layer where the harnessed arm pulled even with the stronger bare model was hygiene, validation, and defect discovery, not the visible feature set — the same detail-execution and review-detectable layer localized in Experiments D and F, now observed on an artifact that actually runs rather than a document that describes one. Experiment G also supplies the program's most concrete counterexample to any claim that the harness is strictly protective: the review loop had to catch a defect that the harnessed arm's own additional architecture introduced. Read together with Experiment E, the honest summary is that review closes gaps it can name and catches defects introduced along the way, but it does not make every underlying technical choice correct, and it does not yet do any of this at a scale or independence level equal to D, F, and E — Experiment G's directional tier and imperfect blinding are real constraints on how far this reading should travel.

9. Limitations

1. **Small n throughout.** Experiment D is n=1 per arm; Experiment F is k=2 per model; Experiment E is a single ceiling run; Experiment G is a single build per arm. None of the four supports a population-level capability claim on its own.
2. **One codebase and one greenfield application.** D and F were run against constraint-rich systems-design tickets in the same repository; G was run against a single greenfield app build. Design or build tasks with different constraint surfaces may behave differently.
3. **Judges drawn from one agent ecosystem, with weaker independence in Experiment G.** In D, E, and F, independence and the blind/sighted split mitigate, but do not eliminate, shared-ecosystem bias in scoring, and that split replicated across four separate instances (Section 3.3, 4.3, 5.2). In G, blinding was imperfect and the operator, also this paper's author, was one of three judges — the weakest independence guarantee in the program, disclosed rather than smoothed over.
4. **Experiment E measures a ceiling, not an average.** A perfect sighted review handed to a repair arm is not representative of typical review quality; it establishes an upper bound on what comprehensive review can close, not a claim about ordinary review passes.
5. **Experiment G does not isolate the harness from build variance.** The bare-Opus and harnessed-Opus arms were separate single builds by different agent instances with different implementation and tooling choices; the harness's contribution is not cleanly separated from ordinary build-to-build or architecture-choice variance.
6. **The un-named origination residual is unmeasured.** No experiment in this program isolates how often review fails to name the thing that actually differentiates a stronger design from a weaker one.
7. **Disclosed procedural deviations.** Two occurred in Experiment D, one in Experiment E's pre-registration timing, and one in Experiment G's judging (partial unblinding via a public commit hash); each is described in place, with its remedy and its assessed impact, rather than omitted.
8. **Fixed order in Experiment D.** Claude Opus 4.8 ran first in every case where order was not explicitly counterbalanced; Experiment F addresses this for the attribution question but does not retroactively de-confound Experiment D itself.
9. **Experiment G's harnessed arm did not use the full available workflow.** It relied on memory, checkpoint, and live-QA discipline, but not on a formal PRD, a design/implementation plan, role adoption, frontend-design tooling, subagent delegation, or a security-scanner pass. The result is a conservative, not a ceiling, estimate of harness uplift.

10. Future work

1. **Replicate Experiment F at a larger k** across additional design tickets and domains to move the attribution result from suggestive toward evidenced.
2. **A named-versus-unnamed origination study**: deliberately withhold the origination-class insight from a repair arm's findings, to measure the un-named residual that Experiment E's disclosed hint transfer could not isolate.
3. **Ordinary-review repair studies**, replacing Experiment E's ceiling review with review quality sampled from the harness's normal operating distribution, to bound how the repair result degrades away from the ceiling.
4. **Cross-domain replication of the four-way zero-memory convergence result**, to test whether constraint-driven architectural convergence (Section 4.3) is a property of this codebase or a more general property of constraint-rich design work.
5. **A standing wire-level model-verification check**, generalizing the mid-run and harvest-time metadata extraction introduced in Experiment F, for any future model comparison run on this harness.

6. **A repair-loop cost study**, quantifying the routing rule in Section 4.4 against actual token and wall-clock cost, to make the architecture/detail-execution split an operational scheduling decision rather than a qualitative recommendation.
7. **A build-variance-controlled replication of Experiment G**: run the same bare-Opus artifact back through the harness, or repeat each of Experiment G's three arms at k>1, to separate harness uplift from ordinary build-to-build and instance variance.
8. **A full-workflow replication of Experiment G**, exercising the broader MindStone capabilities Birch did not use — a formal PRD, a design/implementation plan, role adoption, frontend-design tooling, subagent delegation, and a dedicated security-scanner pass — to test whether the observed uplift margin widens.

11. Conclusion

The question this program set out to answer was narrower than "which model is better at design work," and the answer it produced is correspondingly more useful. Architecture, on constraint-rich work, converges regardless of model tier — four arms with no shared memory whatsoever landed on the same core design in Experiment F, echoing the two-arm convergence already seen in Experiment D. A small, real capability edge does exist, but it lives in execution detail rather than in what gets built, and Experiment F's order-clean, de-anchored design was needed to separate that edge from the second-mover confound that made Experiment D's result ambiguous on its own. That edge is not immune to the review loop: Experiment E showed that a comprehensive sighted review can close it entirely, transporting even an origination-class insight the weaker model did not itself conceive. Experiment G extends the claim from a written artifact to a shipped one, at a lower evidence tier: on a real greenfield build, the same review discipline moved the weaker model into the stronger model's judged quality band by supplying dependency hygiene, validation depth, and defect discovery — while candidly, that same review loop also had to catch a defect the harnessed arm's own added architecture introduced. The boundary this program has not moved is stated plainly: no experiment here measures how often a review fails to name the thing that actually matters, and Experiment G's uplift is not yet isolated from ordinary build variance. Until those residuals are measured, the convergence claim this program supports is real but bounded — evidenced for detection and for repair at a ceiling, suggestive for the size of the underlying design-work gap, directional for shipped-product uplift, and silent on the cases review itself would miss.

Selected references and citation targets

This list is intentionally conservative. Exact bibliographic details and any benchmark comparisons should be verified before formal submission.

- Concepts and conventions of pre-registration in experimental design, as applied to software and agent evaluation.
- Blinded and double-blinded review methodology, including sighted/blind reviewer splits as an instrument rather than merely a bias control.
- Literature on inter-rater reliability and independent-judge convergence as a validity signal.
- Order effects and mover-order confounds in paired comparison and A/B testing designs.
- Prior work on LLM-as-judge and multi-model comparison protocols, including model-serve verification (requested-versus-served identity checks).
- The companion case study establishing harness-level convergence for patterned, reviewable engineering work, and its Event C finding on the limits of review-as-origination.
- Software quality and shipping-readiness evaluation frameworks (dependency hygiene, security scanning, validation depth) as applied to greenfield application benchmarking.

Appendix A: Glossary of terms and abbreviations

Term	Meaning in this paper
Arm	One model's run through the harness on a given ticket, producing one candidate deliverable.
De-anchored arm	An arm run as a fresh subagent instance with no orchestrator memory, recall, or shared state with any other arm.
Design A / Design B	The anonymized labels assigned to Experiment D's two arms after mechanical masking.
Docs A–D	The anonymized labels assigned to Experiment F's four arms after mechanical masking, in sha256 order.
A′	Experiment E's repaired revision of Experiment D's Design A.
Sighted judge (Slate)	The reviewer with access to the actual codebase at the baseline commit, used to check architectural claims against repository reality.
Blind judge (Hearth)	The reviewer working from ticket text and the candidate documents alone, with no repository access.
Five frozen criteria	Constraint correctness, option-space coverage, risk identification, decision quality, and spec completeness; scored 1–5 each across all three experiments.
Defect taxonomy	Design, code-integration, test-oracle, and claim-boundary, the four classes used to categorize judges' docked points.
Combined score	The sum of a document's sighted and blind scores out of 50, used for cross-arm ranking in Experiment F.
Directional / suggestive / evidenced	The three pre-committed evidence tiers used throughout the program, defined in Section 1.3.
Origination-class mechanism	A design element that was not requested by the ticket or any reviewer, and that therefore required the model (or a reviewer's naming of it) to be introduced at all.
Review-recoverable defect	A defect class that repo-grounded review can, in principle and in practice, detect without needing to have originated the correct alternative itself.
Maple / Cedar / Birch	The codenames assigned to Experiment G's three arms: Fable-bare, Opus-bare, and Opus-harnessed respectively.
Fable-bare / Opus-bare	Experiment G arms run in stock Claude Code with no MindStone/LCA review workflow applied.
Opus-harnessed	Experiment G's arm: Claude Opus 4.8 operating under the MindStone/LCA review workflow and

	Hearth's live QA.
Cairn	The agent participant who clarified Experiment G's harness-scope caveat and fixed the Tone.js draw-scheduling defect found by Hearth.
Shipping discipline	This paper's term for the dependency hygiene, security hygiene, validation depth, and defect-discovery layer that Experiment G attributes the harnessed arm's uplift to.

Appendix B: Disclosed protocol deviations

Table 7. All disclosed deviations across the program

Experiment	Deviation	Remedy	Assessed impact
D	Automatic compaction summary re-injected arm-1 design content into arm 2's context before any design work began.	Restarted arm 2 from a fully cleared session with an independently verified, content-clean handoff note.	False start caught before contamination could affect the judged deliverable; no design content reached the completed arm-2 run.
D	The message-relay's 16k-character body cap echoed a rejected whole-document post back into the orchestrating agent's context.	Documents were re-posted in seven chunks each.	Occurred after arm 2 was already frozen and hashed; no experimental impact.
E	The pre-registration was first posted with placeholder commit hashes.	Corrected in an append-only follow-up minutes later.	No effect on the arm, which ran after the correction was posted.
G	Slate's scoring was partially unblinded: Birch's identity was discoverable through a public commit hash (`f0742bb`) before scoring.	Disclosed alongside the results; Mira's independent blind pass and the operator's qualitative judgment are reported alongside Slate's score rather than in place of it.	Scores from Experiment G are reported at the directional tier specifically because of this and the operator's dual role as judge and author.