

Journey-Manifest Failure Gap Matrix and Checklist

Journey-Manifest Failure Gap Matrix and Checklist

Purpose and Audience

The selected workstream is **Internal Knowledge**. This page is for **Lee and future playbook authors** who need a compact way to find an uncovered journey-manifest remedy and identify the operational check that can show whether the remedy is addressed. Its intended outcome is quicker identification of uncovered journey-manifest remedies and required operational checks, while keeping the source boundary explicit and leaving unsupported facts unresolved. The page uses only the three logical sources sealed by `sources/authority.json` and recorded as `public-safe` in `registries/sources.json`.

The sealed authority revision is `2026-08-08`. It names `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`, and sets `employee_may_modify` to `false`. Those records are inputs, not prose to reinterpret. A gap in this matrix means that a remedy, its verification, its owner, or its decision coverage is not explicit in the bounded product artifacts. It does not establish platform behavior, a historical outcome, or any decision that is absent from the sealed inputs.

Failure-Gap Matrix

Use one row per uncovered condition. The owner column names an owner only when the bounded inputs support one; otherwise it records that the decision is missing. A command is a reproducible check to be run by an authorized evaluator, not evidence that it has already run. Evidence references remain limited to the sealed authority and source registry plus the three product artifacts in this step.

Identified failure gap	Practical remedy	Verification command or evidence	Owner	Covered decision or explicit missing decision
The page or journey manifest could drift from the sealed source set, or name a source that is not one of the three authority-listed records.	Compare the page metadata and journey source references with <code>sources/authority.json</code> and <code>registries/sources.json</code> ; remove any reference that is not one of the three <code>public-safe</code> records.	Evidence: the authority <code>source_files</code> array and matching registry records. Command: <code>python3 -B tools/validate_content.py --semantic-only</code> .	Not established in the sealed inputs; do not infer an owner for changing the authority boundary.	Covered by AC-1. The sealed inputs do not record a decision authorizing a fourth source, so no fourth source may be inferred.
The page header and its <code>registries/pages.json</code> entry could disagree on identity, title, visibility, audience, reading modes, claim references, or source references.	Copy the page header values exactly into the single registry entry and keep the claim list empty; then inspect the pair as one synchronization unit.	Evidence: the page metadata block beside the matching registry object. Command: <code>python3 -B tools/validate_content.py --semantic-only</code> when an authorized evaluator is permitted to run the deterministic product check.	Not established in the sealed inputs; do not infer an owner for the bounded synchronization decision.	Covered by AC-1. The matrix records no authority to change the source registry or authority record when synchronization exposes a source decision gap.
A journey can declare a command without showing that the command belongs to the reproducible allowlist or that its trace is covered by an acceptance check.	Keep the allowlisted command in the journey manifest, use only that command in non-exploratory journeys, and make <code>acceptance_checks</code> and <code>traces_to</code> resolve to AC-1, AC-2, or AC-3.	Evidence: the journey manifest's <code>command_allowlist</code> , each journey command, and matching acceptance traces. Command: <code>python3 -B tools/validate_content.py --semantic-only</code> .	Not established in the sealed inputs; the owner of recording an executed result is an explicit missing decision.	Covered by AC-2. The missing decision is who records an executed result when a command declaration is valid but no reader evidence is supplied.
Deterministic product checks, reader evidence, and independent review could be conflated, causing an unexecuted command or a reader statement to be treated as proof.	Label each check by evidence class, retain the exact command and observed result for reader evidence, and have an independent reviewer compare the page, registry, and journey traces without adding facts.	Evidence: a deterministic check result, a reader record with command and output, and an independent comparison of the three bounded artifacts. Command: <code>python3 -B tools/validate_content.py</code>	Not established in the sealed inputs; the evidence owner and reviewer identity remain	Covered by AC-2. No execution result, reviewer identity, or acceptance threshold is

Identified failure gap	Practical remedy	Verification command or evidence	Owner	Covered decision or explicit missing decision
		--semantic-only for the deterministic product check.	explicit missing decisions.	invented by this page.
A failed or uncovered check could lack an explicit next decision, remedy boundary, or owner, leaving authors tempted to fill the gap with unsupported platform or historical claims.	Preserve the gap as unresolved, state the smallest evidence needed to close it, and route a bounded decision that names the owner, allowed evidence, and affected artifact before changing content.	Evidence: the unresolved row itself plus the later decision record, if one is supplied. Do not substitute a guessed result for missing evidence.	Not established in the sealed inputs; Lee is the stated audience for this page, not an inferred owner of a new remedy decision.	Covered by AC-3. Explicit missing decisions remain missing; this page does not convert them into approvals, platform facts, or historical outcomes.

The matrix is complete only when every identified gap has a practical remedy, an executable command or named evidence, an owner or an explicit unresolved owner, and either a covered decision or a clearly recorded missing decision. Adding detail that cannot be tied to the three sealed public-safe sources is a new claim and is outside this page's bounded repair route.

Operational Checklist

Classify the check before recording its result. Deterministic product checks inspect the page, registry, and journey structures and may use only the reproducible commands declared in the journey manifest. Their result speaks to the product artifacts, not to a platform execution. Reader evidence records the exact journey, command, exit status, observed output, source references, and acceptance trace supplied by the reader; a command declaration alone is not reader evidence. Independent review is a separate comparison by a reader who checks the matrix rows, metadata synchronization, source boundary, and AC-1 through AC-3 traces without silently resolving a missing decision.

Use this sequence for each gap:

- Deterministic product checks:** where an authorized evaluator is permitted to run it, use the exact allowlisted command `python3 -B tools/validate_content.py --semantic-only` for the bounded product contract check. Record the command and result separately from interpretation; the command declaration itself is not an execution result.
- Reader evidence:** record the journey ID, exact allowlisted command, exit status, observed output, source or artifact references, and the AC-1, AC-2, or AC-3 trace. If a field is unavailable, mark it unresolved rather than replacing it with an inferred result.
- Independent review:** compare the page header with `registries/pages.json`, compare every journey command with its allowlist, confirm that the three authority-listed sources are the only source references, and verify that every non-exploratory journey traces only to AC-1, AC-2, or AC-3.

For the uncovered remedy selected from a matrix row, verify these evidence fields together: the identified failure gap, practical remedy, exact allowlisted verification command or named evidence, owner or explicitly unresolved owner, and acceptance trace. A reader may verify

that each field is present in the row and its journey trace; the execution status and observed output remain unresolved until a separate reader record supplies them.

The checklist is operationally complete when these three evidence classes are visibly distinct and every gap row points to the evidence class that can verify its remedy. A review may identify a missing decision, but it must not silently supply one.

Missing Decision Escalation

Escalate only the unresolved decision that prevents a row from being closed. The sealed inputs do not supply a new source, a remedy owner, an independent reviewer, an execution result, or an acceptance threshold beyond the bounded records described here. Therefore, an author must not choose one of those facts by reading intent into a locator, a command string, or an empty claim list. Mark the field as a missing decision and preserve the exact gap.

A useful escalation record names the affected matrix row, the three sealed source references consulted, the exact allowlisted command or evidence needed, the artifact to be checked, the proposed decision owner, and the acceptance criterion that would close the row. Until that record exists, keep the remedy unassigned or unresolved and make no content change that would imply a source addition, a platform result, a historical result, or a decision that the sealed inputs do not contain. The smallest safe follow-up is to obtain the missing bounded decision or evidence, then update only the affected row and its synchronized page or journey artifact.

If the missing field is ownership, ask who is authorized to record the deterministic result, reader evidence, or independent review. If it is evidence, ask which exact command or artifact is acceptable. If it is decision coverage, ask which of AC-1, AC-2, or AC-3 closes the row. These questions keep the escalation factual and prevent an unresolved product gap from being recast as platform behavior or a historical outcome.

Acceptance Checks

- **AC-1 — Matrix coverage:** Every identified failure gap has a practical remedy, a verification command or named evidence, an owner or explicitly unresolved owner, and either a covered decision or an explicit missing decision. The row references remain within the three sealed public-safe source records and the bounded product artifacts.
- **AC-2 — Executable operational checks:** The checklist distinguishes deterministic product checks, reader evidence, and independent review; each selected remedy names its failure gap, practical remedy, exact allowlisted verification command or named evidence, owner status, and acceptance trace; every non-exploratory journey uses a command from the reproducible allowlist; and no declared command is presented as execution evidence without a recorded result.
- **AC-3 — Explicit missing-decision handling:** Any absent owner, evidence, reviewer, threshold, or remedy decision remains marked unresolved and is routed with the exact gap, source references, artifact, command or evidence, and proposed decision needed for closure. No platform behavior, approval, publication fact, or historical outcome is inferred.

Provenance: registry-listed (no candidate packet)

Registry Source: `content/journey-manifest-failure-gap-matrix.md`