DOCUMENTATION PAGE

Failure-Triage Decision Guide

Failure-Triage Decision Guide

Purpose

This is an Internal Knowledge guide for auto-orch operators and workflow maintainers. The reader outcome is to consistently classify strict-lint and evidence-chain failures, route each to the correct owner, reject unsupported workarounds, and preserve complete escalation evidence. The sealed authority map records revision `2026-08-08`, exactly three logical source IDs, and `employee_may_modify: false`; each registered source is public-safe. Verify those metadata facts before classifying an observation, but treat them as a bounded source boundary rather than as evidence of a runtime event or a cause. They do not expose the referenced source corpora or runtime evidence, and they do not grant authority to change governance, routing, validators, activation, approval, publication, promotion, rollback, scheduling, or a live directory.

Classify the Failure

Start with the observed record and its exact location, not with a guessed cause. First verify that `sources/authority.json` and `registries/sources.json` still identify the sealed set: `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`, with public-safe disclosure. Use the strict-lint label only when the supplied observation identifies a deterministic product-input problem such as page structure, metadata, a registered reference, or another declared content check. Record the failing path, rule, and actual result; do not turn a missing observation into a lint failure.

Use the evidence-chain label when the supplied record concerns the linkage or completeness of an escalation record, such as its source mapping, command claim, status, observed output, or acceptance trace. This is a record-quality classification, not proof of a product defect. Keep it separate from an

environment or platform fact: runtime behavior and platform attestations are not established by a product-side record. An absent, conflicting, or unsealed item remains unresolved until its owning boundary supplies it. A category describes only the observed gap; it is not permission to infer a cause, repair target, owner, retry, or terminal outcome.

Route the Owner

Route a strict-lint finding to the product-side content or workflow maintainer when the evidence points to a Markdown page, page metadata, registry entry, source reference, or deterministic projection input. The maintainer may repair the declared product source and retain the original observation, but may not rewrite the sealed authority or source registry to make the result pass. Route an evidence-chain finding about runtime execution, evaluator evidence, validator authority, routing, identity, or other platform attestation to the owning evaluator or platform boundary. These are separate routes: do not convert a product-content defect into a platform attestation, or a missing platform attestation into a product defect. A workflow maintainer may package the product-side handoff and reconcile its references; that does not certify the external evidence.

The sealed map names ownership boundaries, not a person, queue, or new routing authority. If the supplied evidence names no owner, record exactly `owner unresolved`, preserve the evidence, and escalate through the governed route already responsible for that boundary. Do not choose an owner from a filename, a generated projection, an inferred source, or an unregistered runtime claim. A successful local check can support a product conclusion, but cannot be used as a platform attestation or as permission to cross a separate control boundary.

Preserve Escalation Evidence

Keep the original observation beside every later result; never replace the first failure with a later pass. The escalation record should name the journey or work item, the exact non-empty command claimed by the manifest, the exit status, the observed output or mismatch, the relevant page or source path, the source references, and the `traces_to` acceptance checks. For this guide the declared command is exactly `PYTHONDONTWRITEBYTECODE=1 python3 -m pytest`; its presence in an allowlist is only a re-executable product claim, not proof that a platform executed it. Record whether each observation is strict-lint, evidence-chain, or unresolved, and identify the owner boundary without filling unknown fields with prose.

Preserve the sealed authority revision, the three logical source IDs, their public-safe status, and the `employee_may_modify` value as metadata. Do not copy private governance material, private source corpora, or runtime evidence into the page. Reject an unsupported workaround, including changing a source, claim, command, or authority record merely to silence a mismatch. Do not hand-edit `build/`, or treat a missing attestation as a failure owned by product content. A complete escalation makes the gap auditable and reversible while leaving publication, promotion, scheduling, rollback, and live-directory state untouched.

Acceptance Checks

- **AC-1 — Classify from sealed inputs:** The reader verifies authority revision `2026-08-08`, the exact three logical source IDs, public-safe registration, and `employee_may_modify: false`; then the reader labels only an observed deterministic product-input mismatch as strict-lint and leaves missing, stale, conflicting, or runtime-only facts unresolved.
- **AC-2 — Route by ownership boundary:** The reader routes a product-side strict-lint finding to the content or workflow maintainer, routes runtime or platform-attestation evidence to its owning evaluator or platform boundary, and records `owner unresolved` when the supplied evidence names no owner; the reader does not invent a person, queue, or new authority.
- **AC-3 — Preserve a complete escalation:** The reader retains the original failure beside later results with the journey or work item, exact command, status, observed output, paths, source references, and `traces_to` values, rejects unsupported workarounds, and distinguishes product conclusions from platform-owned execution or validator attestations.

Provenance: registry-listed (no candidate packet)
Registry Source: `content/failure-triage-decision-guide.md`