

Regression Discipline

Regression discipline is how GROW prevents the most common operational failure: a "small change" that quietly breaks a previously-passing behavior. This playbook governs the comparison harness, the evaluation change log, intentional-regression declarations, and the release gate.

When Regression Runs Fire

A regression run is required on any of the following changes to a GROW system:

- Model swap (provider, family, version, or routing tier).
- Prompt change in any artifact tagged `prompt_critical`.
- Tool change — addition, removal, signature change, or backend swap.
- Retrieval index rebuild or corpus change.
- Workflow change — added step, reordered steps, changed HITL gate, changed fallback path.
- Schema change to `s2-audit-trail-schema` or `s3-provenance-metadata-schema`.

Cadence regressions also fire weekly against the frozen reference, regardless of changes.

Comparison Harness

The harness runs the same test suite against three artifacts in lockstep:

1. Champion — current production-tagged bundle.
2. Challenger — the candidate.
3. Frozen reference — a permanently tagged historical bundle for drift detection.

For each test, the harness collects: verdict, scorecard fragment from `s2-scoring-system`, provenance record from `s2-audit-trail-schema`, and a structural diff of `decision_trace` against champion.

Harness Output

The harness emits a regression report with:

- Per-test triple: champion verdict, challenger verdict, frozen-reference verdict.
- Per-family aggregate scores.
- Drift indicators carried forward from `s2-scoring-system`.
- A delta classification for each test: improved | unchanged | intentional-regression | unintentional-regression | new-fail.

Delta Classifications

- improved — challenger ties or beats champion; no human action required.
- unchanged — verdicts and scores within ± 0.2 weighted average on the same test.
- intentional-regression — challenger fails or scores lower, AND a declaration is on file (see below).
- unintentional-regression — challenger fails or scores lower without a declaration. Blocks release.
- new-fail — challenger fails a test that did not exist when champion was tagged. Treated as unintentional-regression until reviewed.

Evaluation Change Log

Every regression run produces one change-log entry, appended (never edited) to the module's change log. Schema:

```
- entry_id: ulid
  created_at: timestamp
  author: actor          # e.g., builder:doug.liles
  change_summary: string  # one sentence
  change_kind: enum(model|prompt|tool|retrieval|workflow|schema|cadence-review)
  artifacts_touched: [artifact_id, ...]
  champion_version: semver
  challenger_version: semver
  regression_report_id: string
  delta_summary:
    improved: int
    unchanged: int
    intentional_regression: int
    unintentional_regression: int
    new_fail: int
  intentional_regression_refs: [test_id, ...] # must be empty if none declared
  evaluator_signatures: [ {role, actor, verdict, signed_at}, ... ]
  decision: enum(promoted|blocked|rolled-back|deferred)
  notes: string
```

Entries with decision = promoted and any unintentional_regression > 0 are rejected by the gate validator.

Example Entries

```
- entry_id: 01HZ4...
  author: builder:doug.liles
  change_summary: "Permit triage prompt – added Walton-only jurisdictional preamble."
  change_kind: prompt
  artifacts_touched: [permit-triage-prompt]
  champion_version: 0.4.1
  challenger_version: 0.4.2
  delta_summary: { improved: 2, unchanged: 41, intentional_regression: 0,
unintentional_regression: 0, new_fail: 0 }
  evaluator_signatures:
    - { role: builder-self-test, actor: builder:doug.liles, verdict: pass, signed_at: 2026-
05-26T15:10Z }
    - { role: domain-expert-reviewer, actor: evaluator:m.tanaka, verdict: pass, signed_at:
2026-05-26T16:02Z }
  decision: promoted

- entry_id: 01HZ5...
  author: builder:doug.liles
  change_summary: "STR reconcile – swapped model tier to cheaper router."
  change_kind: model
  champion_version: 0.7.3
  challenger_version: 0.8.0
```

```
delta_summary: { improved: 0, unchanged: 38, intentional_regression: 1,
unintentional_regression: 0, new_fail: 0 }
intentional_regression_refs: [Q-02]
evaluator_signatures:
  - { role: human-evaluator, actor: evaluator:doug.liles, verdict: pass-with-caveat,
signed_at: 2026-05-27T10:14Z }
decision: promoted
notes: "Tone score on Q-02 dropped 4→3 (still above floor). Cost reduction 38%. Accepted
as intentional regression with monthly review."
```

Declaring an Intentional Regression

A scored drop, an added refusal, or a tightened HITL gate may be intended. Declaration rules:

1. Pre-declared, not post-rationalized. The declaration must be filed BEFORE the harness runs the regression, in a regression_intent.yaml PR-attached file. Post-hoc declarations are rejected.
2. Per-test specificity. Each declared regression names the test_id and the expected change (e.g., "Q-02 will drop one point on tone score; pass threshold still met").
3. No floor violations. An intentional regression may not push a criterion below its severity floor from s2-scoring-system. Critical floors are never waivable.
4. No safety regressions, ever. A declared regression in the safety family is a contradiction in terms and is rejected by the gate validator.
5. Time-boxed. Every intentional regression has an expires_at (default 90 days). At expiry, the regression must be retired (resolved) or re-declared.

Release Gate on Regression Results

A challenger may be promoted only when ALL of the following hold:

- delta_summary.unintentional_regression == 0 and new_fail == 0.
- Every declared intentional regression has a matching pre-declared regression_intent.yaml entry.
- All severity floors honored per s2-scoring-system.
- Drift indicators are not in a blocking state (two simultaneous = block).
- Evaluator signatures present per s2-evaluator-roster invocation rules for the system's tier.
- The change-log entry is written, signed, and committed before traffic shifts.

The gate is enforced by tooling, not by trust. A builder who bypasses the gate creates an audit-trail anomaly that the external-auditor role will surface.

Rollback

Rollback is a first-class action, not a failure. A decision = rolled-back entry is written within 1 hour of detecting a regression in production traffic. Rollback restores the prior champion bundle and opens a parent investigation entry.