# Pre-Merge Behavioral Analysis for Prompt Revisions

mowa

TECHNICAL DESIGN NOTE · REVIEW WORKFLOW

**ABSTRACT**

*Prompt revisions in production large language model systems are often promoted to release without a structured behavioral review step. This note describes the pre-merge analysis workflow implemented by mowa, in which every candidate prompt revision is scored, structurally diff'd against the prior version, and optionally tested against a stored input set before a pull request is opened. The output of this analysis is surfaced both inside the editor and on the pull request itself, where it acts as advisory context for human reviewers. The current implementation does not gate merge; the GitHub merge button remains the source of truth for release. We describe the analysis surface that exists, the deliberate decision to keep gating advisory rather than enforced, and the steps required to evolve toward enforceable checks where customers want them.*

## Problem

A prompt is a configuration object that can change observable system behavior as dramatically as a code change can, yet the review workflow around prompt edits is usually weaker than the workflow around code. A code change goes through a typed compiler, a test suite, and a reviewer who reads a diff. A prompt change is often a string commit with a one-line message.

mowa addresses this asymmetry by attaching a structured pre-merge analysis to every prompt edit, making the behavioral change visible at the review surface rather than leaving it implicit in the textual diff alone.

## Analysis Pipeline

When a user opens the editor and modifies prompt content, the change is held as a draft in browser storage and in a per-user server-side draft slot. Drafts do not advance the current pointer; the prompt is unchanged from the perspective of any downstream consumer until the user explicitly elects to send the change for review.

On request, the analyzer runs four passes over the draft, in this order.

- **Health re-score.** The LLM auditor re-evaluates the draft against the same rubric used at scan time and emits a new health score plus a structured issue list. The score is shown next to the previous version's score so the delta is immediately visible.
- **Structural diff.** The editor renders a token-level diff between the prior live version and the draft, highlighting additions, removals, and modified spans.
- **AI-suggested rewrites.** An optional pass generates a small number of alternative phrasings that address issues found by the auditor, each tagged with a short rationale. The user can accept or dismiss.
- **Test runs.** If the prompt has a stored test-input set, the draft can be run against the inputs and the responses scored. The results are stored against the draft and shown in the Test tab as a comparison strip.

## Pull Request Surface

When the user elects to ship the draft, mowa opens a pull request against the source repository. The PR body is generated automatically from the diff summary and includes a machine-readable footer with the prompt name, file path, before-score, after-score, and a link back to the analysis view in mowa. The intent is that a reviewer who knows nothing about mowa can still understand what changed and why from the PR body alone.

The PR comment is advisory. mowa does not block merge, set a required status check, or interpose between the reviewer and the GitHub merge button. The user who can merge a code change to the same repository can merge this PR with the same authority.

## Why Advisory Rather Than Enforced

Enforced behavioral gating is technically feasible — a GitHub status check could be posted reflecting pass/warn/fail derived from the analysis outputs — and is on the roadmap as an opt-in. It is not the default for two reasons.

- **Behavioral metrics are noisy.** Single-run health scores and test-input responses both carry sampling variance. Enforcing a threshold against a noisy signal converts product friction into infrastructure friction without proportional behavioral gain. Soft surfacing lets the human apply judgment.
- **Trust is earned per workspace.** A workspace that has accumulated several months of test-input outcomes and tuned its score threshold against real regressions is in a different position than a workspace on its first prompt. Enforcement is the right end-state for the first workspace and the wrong default for the second.

The PR comment carries the same data a future status check would carry; switching from advisory to enforced is a configuration decision, not a redesign.

## Status and Limits

The pre-merge analysis pipeline is implemented for every tracked prompt and runs on demand from the editor. The PR comment is posted on every PR mowa opens. The optional GitHub status check, configurable thresholds, and required-review enforcement are designed but not yet shipped. Behavioral dimensions richer than the single integer health score (refusal rate, output-length distribution, embedding-based semantic consistency) depend on the behavioral fingerprint extension described in the companion note on the prompt provenance registry.